

Universidad Carlos III de Madrid

Escuela Politécnica Superior



Grado en Ingeniería de Sistemas Audiovisuales

Trabajo Fin de Grado

Simulador Flash para gamificar historias interactivas

Autor: Esther Varas García

Tutor: M^a Carmen Fernández Panadero

Septiembre 2013

Peor que ver la realidad negra, es el no verla

(Antonio Machado)

Agradecimientos

Agradecer a toda la gente que ha estado a mi lado durante estos años de formación, que me han apoyado y ayudado a lograr mi meta.

Primeramente dar las gracias a mi familia que ha hecho posible que yo pueda hacer lo que me gusta.

Después agradecer a todas mis amigas que me hayan aguantado todos estos años haciéndome pasar grandes momentos.

Gracias a todas esas personas que me he ido encontrando a lo largo de estos últimos años en la universidad, que pasaron de ser compañeras de clase a buenas amigas. Esos momentos de risas en las clases y laboratorios nunca podrán ser olvidados, gracias a esos momentos ha merecido la pena pasar tantas penurias en estos años.

También agradecer a la gente que he conocido en Vodafone por ayudarme a completar mi formación. A mis compañeras becarias, que sin ellas las mañanas se harían muy largas.

Y sobre todo gracias a M. Carmen Fernández Panadero por ofrecerme un proyecto tan interesante. Gracias por esos ánimos en los momentos más duros de este proyecto y su paciencia en la creación de esta memoria. Y sobre todo gracias por haber sido más que una tutora.

Gracias a todos por haberme ayudado a ser quién soy.

Resumen

A lo largo de los años, los simuladores han demostrado ser una herramienta eficaz para el aprendizaje. Parte de su éxito se debe a la posibilidad de recrear escenarios de diferente naturaleza, a las posibilidades que ofrecen al alumno de aprender de sus errores en un entorno controlado, y el papel activo que otorgan al estudiante en su propio aprendizaje. Sin embargo, una de las mayores dificultades para su implantación es que se trata de una herramienta compleja, que requiere un equipo pluridisciplinar para su desarrollo y que en general resultan excesivamente caros no sólo económicamente sino también en tiempo. La creación de un simulador requiere de un gran esfuerzo de diseño, desarrollo y despliegue, este problema se agrava en el contexto económico actual en el que por un lado la crisis económica dificulta la inversión en educación y por otro el aumento de ratio de los alumnos por profesor en las aulas demanda herramientas que faciliten el aprendizaje autónomo.

La finalidad de este trabajo es la creación de un meta-simulador educativo en 2D que simplifique el proceso de desarrollo. Se trata de un simulador que guía a una persona sin conocimientos de programación para diseñar sus propios juegos o simuladores que ayuden al aprendizaje. El simulador facilita al educador tanto la introducción de los elementos del juego como escenarios, personajes y tesoros, como en la creación de los ficheros que formarán parte de la herramienta final que posteriormente entregará a sus alumnos para dar soporte a su aprendizaje.

El simulador está basado en una herramienta previa llamada SimufLOW que proporciona una biblioteca básica de elementos gráficos desarrollados en flash para la presentación de contenidos. Esta herramienta se ha extendido para introducir elementos de juego y storytelling que permitan guiar al usuario en el desarrollo de un simulador completo.

Se ha tratado de elaborar una herramienta barata, rápida y sencilla de manejar que permita a un usuario sin conocimientos de programación desarrollar un juego desde cero. El resultado es una herramienta flexible y multifuncional que permite el desarrollo de simuladores para distintas franjas de edad y distintas necesidades como por ejemplo la docencia de una materia concreta para personas de cualquier edad, ayudar al establecimiento de rutinas en un entorno conocido a personas de avanzada edad o para la creación de cuentos personalizados para niños de corta edad.

Palabras clave: simulador, e-learning, storytelling, narrativa, juegos educativos

Abstract

During the last years, simulators have proven to be an effective learning tool. Part of their success is due to their ability to recreate different scenarios, to offer students opportunities to learn from their mistakes in a controlled environment, and to the active role assigned to the students in their own learning. However, a major difficulty for implementation is that it is a complex tool that requires a multidisciplinary team to develop, and that they are, in general too expensive, not only economically but also in time. The creation of a simulator requires a great effort of design, development and deployment. These problems are exacerbated, in the current economic context in which on one hand the financial crisis becomes more difficult investment in education, and other increased ratio of student per teacher in the classroom demand tools that facilitate autonomous learning.

The purpose of this work is to create a meta-simulator in 2D that simplifies the development process. This is a simulator that guides a person without programming skills to design their own games or simulators that support learning. The simulator facilitates the educator both: the insertion of game elements such as stages, characters and treasures, and creating files that will be part of the final tool, later given to the students to support his learning.

The simulator is based in a previous tool called SimufLOW that provides a basic library of graphic elements developed in flash for content presentation. This tool has been improved to insert game elements and storytelling to help guide the user through the development of a complete simulator.

In this project we have tried to develop an inexpensive tool, fast and easy to use that allows a user with no programming skills to develop a game from scratch. The result is a flexible and multifunctional tool that allows the development of simulators for different age groups and different needs such as the teaching of a particular subject for people of different ages, helping to establish routines in familiar surroundings for people of advanced age, or for creating personalized stories for young children.

Keywords: simulator, e-learning, storytelling, narrative, educational games

Índice general

1.Introducción y Objetivos	1
1.1 Introducción	1
1.2 Motivación	3
1.3 Objetivos	4
1.3.1 Objetivos funcionales.....	4
1.3.2 Objetivos técnicos	5
1.4 Estructura de la memoria.....	5
2.Estado del Arte	6
2.1 Análisis del Estado del Arte	6
2.1.1 e-Adventure	6
2.1.2 Storybird.....	10
2.1.3 Conclusiones del Estado del Arte	15
2.2 Estándares empleados	17
2.2.1 XML.....	17
2.2.2 SCXML.....	17
2.2.3 HTML	17
2.2.4 JavaScript.....	18
2.2.5 MySQL	18
2.3 Herramientas necesarias.....	18
2.3.1 Simuflow.....	19
2.3.2 XAMPP.....	19
2.3.3 Adobe Flash Player	21
2.3.4 Navegador Firefox Mozilla	21
3.Diseño y Arquitectura del Sistema	22
3.1 Descripción general.....	22
3.2 Requisitos funcionales y técnicos.....	23
3.2.1 Requisitos funcionales.....	23
3.2.2 Requisitos técnicos.....	23
3.3 Diseño. Elementos gráficos	24
3.3.1 Elementos gráficos de SimGame.....	24
3.3.2 Juego de clasificar	25
3.3.3 Juego de ordenar.....	26

3.2.3 Ventanas de formulario.....	26
3.4 Diseño. Elementos del simulador.....	27
3.5 Diseño. Arquitectura	34
3.7 Diseño. Modelo de datos	37
3.7.1 Ficheros XML y SCXML	37
3.7.2 Base de datos MySQL.....	40
3.7.3 Ficheros personalizados del juego del usuario	45
4.Validación y Pruebas	47
5.Planificación y presupuesto	49
5.1 Planificación	49
5.2 Presupuesto	52
5.2.1 Coste personal.....	52
5.2.2 Coste material	52
5.2.3 Resumen de los costes	53
6.Líneas futuras y conclusiones.....	54
6.1 Futuros desarrollos del simulador.....	54
6.1.1 Pasar de Flash a HTML5.....	54
6.1.2 Crear archivo zip con el juego creado	54
6.1.3 Agregar lógica para crear videojuegos de aventuras	54
6.1.4 Marco regulador.....	55
6.2 Conclusiones.....	55
Anexo 1. Escenario creado con XML	58
Anexo 2. Fichero flujo creado con SCXML.....	59
Anexo 3. Formulario creado con HTML.....	60
Anexo 4. Guía de usuario	61

Índice de figuras

Imagen 1. Elementos interacción en rojo. Barra seguimiento en azul. Ayuda en verde	2
Imagen 2. Cuadro de diálogo.	3
Imagen 3. Ventana emergente.	3
Imagen 4. Pantalla edición e-Adventure Rojo menú derecho, verde menú superior.	7
Imagen 5. Crear escena. Rojo ventana imagen fondo, azul información, verde transición escenas.....	8
Imagen 6. Crear objeto. Rojo imagen, azul información y verde acciones.....	8
Imagen 7. Crear personaje. Rojo imagen, azul elegir presentación texto y verde acciones con personaje.....	9
Imagen 8. Crear conversación. Rojo crear nodos y azul introducir textos.	9
Imagen 9. Crear libro. Rojo introducir texto, azul elegir apariencia.	10
Imagen 10. Pantalla registro Storybird. Rojo perfiles de usuario.	11
Imagen 11. Pantalla creación Storybird. Rojo menú derecho temas.....	11
Imagen 12. Pantalla elección historia Storybird. Rojo tipos de historia.	12
Imagen 13. Pantalla crear cuento. Rojo línea con las hojas del cuento,.....	12
Imagen 14. Editar imagen.	13
Imagen 15. Editar portada. Rojo tipos portadas y azul editar título.....	14
Imagen 16. Pantalla edición poema. Rojo colores texto, azul cuadro editar poema y verde textos.....	14
Imagen 17. Editar texto.....	15
Imagen 18. Componentes gráficos de Simuflow: (a) ActionPoint, (b) UserActionPanel,	19
Imagen 19. Pantalla instalación XAMPP.....	20
Imagen 20. Panel de control XAMPP.....	20
Imagen 21. Web 127.0.0.1. Fuente: 127.0.0.1.....	21
Imagen 22. Diagrama casos de uso SimGame. En el sistema actual hay una única base de datos para almacenar información de los educadores y una base de datos por cada nuevo simulador creado por el usuario final, para almacenar la información de sus alumnos. Estas bases de datos se pueden distribuir para almacenar por separado diferentes tipos de información.	22
Imagen 23. Formulario juego clasificar.	25
Imagen 24. Formulario juego ordenar.	26
Imagen 25. Formulario crear formulario con número inputs variable.	27
Imagen 26. Formulario para introducir el número de elementos en el juego.....	28
Imagen 27. Pantalla principal que da acceso a todos los elementos del juego a)terrenos, b)objetos,	29
Imagen 28. Pantalla ayuda simulador. Rojo explicación elemento,	29
Imagen 29. Módulos del simulador SimGame. Módulos: core, HTML y lib.....	35
Imagen 30. Diagrama flujo de SimGame.....	36
Imagen 31. Fichero configuración SimGame	37
Imagen 32. Código XML pantalla principal SimGame	38
Imagen 33. Ejemplo creación terreno juego. a) formulario, b) código XML, c) resultado.....	39
Imagen 34. Fichero flujo SCXML. Recuadro rojo estado, azul transición y verde evento.....	39
Imagen 35. Ejemplo flujo SCXML. a) formulario presentado en SimGame para introducir el flujo y b) código generado para los ficheros personalizados del usuario.	40

Imagen 36. Ejemplo guardar terreno BD. a) Arrays en la BD, b) código HTML formulario	41
Imagen 37. Ejemplo menú elegir número elemento a crear	41
Imagen 38. Ejemplo sacar datos objeto BD. a) arrays BD b) inputs formulario editar objeto....	42
Imagen 39. Ejemplo menú editar elemento objeto para saber posición array	42
Imagen 40. Ejemplo crear elemento formulario. a) array creados en la BD, b) formulario mostrado al usuario para crear formulario	43
Imagen 41. Ejemplo crear mini-juego. a) formularios html del reto, b) arrays en la BD	44
Imagen 42. Ejemplo crear flujo juego. a) arrays BD y b) formulario de flujo.....	44
Imagen 43. Ejemplo escribir ficheros personalizados. a) código JavaScript para crear fichero b) arrays flujo guardados en BD	46
Imagen 44. Guía usuario. Panel control XAMPP	61
Imagen 45. Guía usuario. panel phpMyAdmin.....	62
Imagen 46. Guía de usuario. Crear base de datos	62
Imagen 47. Guía de usuario. Importar fichero crear tablas BD	63
Imagen 48. Guía de usuario. Tablas creadas en la BD.....	63
Imagen 49. Guía de usuario. Insertar usuario y contraseña	64
Imagen 50. Guía de usuario. Ejemplo usuario creado en la BD	64
Imagen 51. Guía de usuario. Identificar usuario	65
Imagen 52. Guía de usuario. a) terrenos b)objetos c) tesoros d) retos e) flujo.....	65
Imagen 53. Guía de usuario. Guía de usuario	66
Imagen 54. Guía de usuario. Introducir número de terrenos, objetos, retos y tesoros	66
Imagen 55. Guía de usuario. ActionPanel terrenos	67
Imagen 56. Guía de usuario. a) Crear terreno b) Ver terreno.....	67
Imagen 57. Guía de usuario. Menú elegir terreno a ver	68
Imagen 58. Guía de usuario. Editar terreno.....	68
Imagen 59. Guía de usuario. Crear objeto	69
Imagen 60. Guía de usuario. Editar objeto	69
Imagen 61. Guía de usuario. Crear presentación del reto	70
Imagen 62. Guía de usuario. Imágenes de ejemplo en el menú de tipo de presentación	70
Imagen 63. Guía de usuario. Elegir tipo de reto	71
Imagen 64. Guía de usuario. Crear juego de clasificar.....	71
Imagen 65. Guía de usuario. Ejemplo juego de clasificar	72
Imagen 66. Guía de usuario. Crear un elemento formulario a) interfaz formulario b) Crear formulario	72
Imagen 67. Guía de usuario. Crear flujo del juego.....	73
Imagen 68. Guía de usuario. Manual crear ficheros personalizados del juego	74
Imagen 69. Guía de usuario. Crear ficheros del terreno del juego	74
Imagen 70. Guía de usuario. Ejemplo guardar fichero juego.....	75
Imagen 71. Guía de usuario. Crear fichero de configuración del juego.....	75
Imagen 72. Guía de usuario. Borrar tranking BD	76
Imagen 73. Guía de usuario. Ejemplo simuladores creados	76

Índice de Tablas

Tabla 1. Objetivos funcionales del proyecto	4
Tabla 2. Objetivos técnicos del proyecto	5
Tabla 3. Conclusiones Estado del Arte	16
Tabla 4. Requisitos funcionales simulador SimGame	23
Tabla 5. Requisitos técnicos del simulador SimGame.....	23
Tabla 6. Descripción elemento terrenos del simulador SimGame.....	30
Tabla 7. Descripción elemento objetos del simulador SimGame	30
Tabla 8. Descripción elemento tesoros del simulador SimGame	31
Tabla 9. Descripción elemento retos del simulador SimGame	31
Tabla 10. Descripción elemento formularios del simulador SimGame.....	32
Tabla 11. Descripción elemento probar mini-juegos del simulador SimGame.....	32
Tabla 12. Descripción elemento ayuda del simulador SimGame.....	33
Tabla 13. Descripción elemento flujo del simulador SimGame	33
Tabla 14. Descripción del elemento crear ficheros del juego del simulador SimGame.....	33
Tabla 15. Compatibilidad del simulador SimGame en navegadores.....	47
Tabla 16. Duración de las tareas del proyecto. Al principio con F más un número se especifica en qué fase de las comentadas anteriormente se encuentra cada tarea	51
Tabla 17. Coste personal proyecto.....	52
Tabla 18. Coste de material del proyecto	52
Tabla 19. Resumen costes proyecto.....	53

Capítulo 1

Introducción y Objetivos

1.1 Introducción

En los últimos años el uso de simuladores como herramienta complementaria al aprendizaje convencional ha ido en aumento [8]. Esto se debe en parte a su interés educativo y su flexibilidad. Los simuladores sirven para recrear una situación en la que puede intervenir un alumno en un entorno controlado, aunando en un mismo recurso educativo información teórica y una componente de práctica y permitiendo a los alumnos probar y fallar sin ningún tipo de peligro [18].

En la actualidad, los simuladores están siendo utilizados con éxito para diferentes perfiles de usuarios y en diversos sectores de negocio como en la aviación [29], medicina [20], emprendimiento [11], la formación laboral [21], o la educación convencional [14].

Los simuladores educativos presentan innumerables ventajas, la Asociación Española de Distribuidores y Editores de Software de Entretenimiento [30] respalda que los juegos son un estimulante método de enseñanza haciendo que la experiencia sea más divertida y entretenida, aumentando la motivación y la concentración por parte del alumno para superar los retos que se le presentan.

Otra gran ventaja es que el aprendizaje está centrado en el alumno, haciéndoles responsables de su propio aprendizaje y los tiempos que se marca cada alumno para aprender proporciona la ventaja de flexibilidad en el aprendizaje [7].

Las ventajas de experiencia divertida y flexibilidad en el aprendizaje posibilitan que el aprendizaje por simulación sea más eficaz ya que se aprende más en menos tiempo.

Una ventaja importante que proporciona el uso de simuladores es que se trabaja en un entorno controlado, por tanto se puede realizar cualquier tipo de práctica sin correr ningún peligro [19].

En el caso de los simuladores de formación laboral presentan innumerables ventajas como flexibilidad horaria, ahorro de costes, disminución de las horas dedicadas a la formación. Todo esto se consigue porque con el uso de simuladores se evitan desplazamientos, suelen ser más cortos en el tiempo lo que reduce que el empleado pierda muchas horas laborables en formación, además el trabajador puede decidir hacer el curso cuando la carga de trabajo sea menor.

El uso de simuladores en el aprendizaje tiene la gran ventaja que pueden ser usados por cualquier persona, no están limitados por edad, en el mercado hay simuladores de ocio para niños de corta edad, simuladores más profesionales para empresas o simuladores para ayudar a personas mayores en su día a día [23].

Por último la ventaja que hace que los simuladores cada vez sean más usados en el ámbito empresarial es el gran número de simuladores para distintas líneas de negocio, hoy en día hay simuladores para casi cualquier práctica médica como simulación quirúrgica, cirugía laparoscópica [9]. Simuladores medioambientales [17]. Simuladores para la de operadores de fábrica [13].

Para ver la estructura que suelen presentar los simuladores en 2D vamos a presentar el simulador empresarial de la Comunidad de Madrid [11]. Todos los simuladores presentan una estructura similar con un escenario principal donde el usuario tiene varios elementos seleccionables para ir moviéndose a través del simulador e ir superando los retos. Además suelen contener una barra de estado para seguir la evolución del alumno y botones de ayuda, todos estos elementos están destacados en la Imagen 1.



Imagen 1. Elementos interacción en rojo. Barra seguimiento en azul. Ayuda en verde
Fuente: captura del simulador empresarial de la CM [11]

Para la interacción del alumno con los personajes del simulador se suele utilizar cuadros de diálogo como se puede ver en la Imagen 2.



Imagen 2. Cuadro de diálogo.
Fuente: captura simulador empresarial de la CM [11]

Otro elemento utilizado para proporcionar ayuda e información al usuario son las ventanas emergentes como se ve en la Imagen 3.



Imagen 3. Ventana emergente.
Fuente: captura simulador empresarial de la CM [11]

Algunos de los elementos comunes en los simuladores analizados serán incorporados en el simulador que se desarrolla con este proyecto.

1.2 Motivación

Las motivaciones que han impulsado a la creación de este proyecto han sido varias, la primera es debido al aumento en el uso de simuladores en los últimos años y la falta de herramientas de autorías orientadas a docentes sin experiencia ni conocimientos en programación.

Por otro lado la motivación en el ámbito económico debido a la recesión en la que se encuentra hoy en día nuestro país dificulta la inversión en educación y en la formación. Para crear un simulador se necesita mucho tiempo en el desarrollo de la herramienta para satisfacer las necesidades del cliente además del desembolso económico que supone la adquisición de un simulador. Es importante desarrollar herramientas que faciliten el desarrollo rápido de estos simuladores para cubrir necesidades puntuales de aprendizaje en el momento en el que se presentan sin necesidad de tener que esperar largos tiempos de desarrollo e importantes inversiones de dinero.

Por último la motivación en el ámbito social ha sido debido a que nos encontramos con una educación reglada en el que el ratio de alumnos por profesor ha aumentado considerablemente, además con la introducción en las universidades del Espacio Europeo de Educación Superior como el fomento del aprendizaje durante toda la vida y la necesidad continua de reciclaje de los profesionales de cualquier sector, exige unas herramientas que fomenten el aprendizaje autónomo y centrado en el alumno.

1.3 Objetivos

El objetivo global de este proyecto es crear una herramienta que permita la creación de juegos y simuladores personalizados de una forma sencilla, rápida, versátil e intuitiva.

Además en el marco socio-económico, debemos de tener en cuenta a la hora de crear la herramienta la situación de crisis que está atravesando el país, por ello se busca crear una herramienta económica por su versatilidad en la creación de diferentes tipos de simuladores y juegos.

1.3.1 Objetivos funcionales

Tabla 1. Objetivos funcionales del proyecto

Identificador	Descripción
OF - 1	Disminución del tiempo empleado en la creación de juegos y simuladores personalizados
OF - 2	Facilitar a los usuarios la inclusión en el juego de recursos disponibles a su alcance
OF - 3	Facilitar el uso de la herramienta a cualquier perfil de usuario
OF - 4	Facilitar a perfiles no técnicos la creación de juegos

1.3.2 Objetivos técnicos

Tabla 2. Objetivos técnicos del proyecto

Identificador	Descripción
OT - 1	Facilitar la introducción de la información en la base de datos
OT - 2	Personalización del juego con varios tipos de componentes, texto, imágenes, vídeos, etc.
OT - 3	Separación de la introducción de la información necesaria para la creación de un juego, de la creación de los ficheros que componen el juego
OT - 4	Crear una estructura común para todos los juegos creados
OT - 5	Seguimiento de usuario a través del juego y del simulador

1.4 Estructura de la memoria

Esta memoria se divide en capítulos, en este apartado vamos a resumir en qué partes está estructurada la memoria.

Capítulo uno se cuentan los simuladores actualmente en el mercado y un estudio sobre la interfaz gráfica.

Capítulo dos se estudian dos herramientas similares a la que se desarrolla en este proyecto. Además se introducen los estándares y las herramientas utilizadas.

Capítulo tres se presentan las soluciones y el diseño que se han desarrollado en el proyecto.

Capítulo cuatro se describen las pruebas y validaciones que se han realizado en el simulador.

Capítulo cinco se presenta la planificación que se ha seguido en la creación del proyecto y el presupuesto que costaría realizarlo.

Capítulo seis son las conclusiones que sacamos con la realización del proyecto y los siguientes pasos a desarrollar en el simulador.

La última parte del proyecto se encuentra todos los anexos que se pueden consultar para tener más información sobre el simulador.

Capítulo 2

Estado del Arte

2.1 Análisis del Estado del Arte

Aunque los simuladores están en auge, todavía hay escasez de herramientas sencillas que pueda el educador manejar para crear sus propios simuladores. Entre las herramientas de autoría orientadas a la creación de este tipo de experiencias podemos encontrar dos grandes bloques. Por una parte los juegos que integran elementos de gamificación y ofrecen una experiencia más interactiva al usuario, y por otra parte las herramientas para la creación de cuentos o historias personalizadas que son en general más intuitivas y sencillas de utilizar pero menos flexibles porque suelen utilizar estructuras argumentales bastante lineales.

Entre las herramientas para la creación de juegos podemos encontrar e-Adventure [6], GameCreators [24], GameMaker [25], o MissionMaker [15] todas estas herramientas proporcionan utilidades para crear juegos a bajo coste, pero tienen una curva de aprendizaje más compleja.

Entre las herramientas para la creación de narrativas (storytelling) podemos encontrar algunas muy flexibles y de orientación profesional como Conducttr [26] que requiere conocimientos avanzados y otras herramientas más sencillas orientadas al desarrollo personalizado de historias en entornos educativos como ourStory [10] o Storybird [22].

Este proyecto trata de combinar elementos de juego y de narrativa en una misma herramienta, pero este es un proceso complejo porque [5] es difícil de combinar la estructura del juego que es principalmente no lineal para fomentar la interactividad con la estructura narrativa que es principalmente lineal para seguir un argumento.

En este apartado se analiza la interfaz y estructura de una herramienta de cada una de estas dos categorías (1) juego y (2) narrativa, para identificar los elementos relevantes para el diseño de una solución que combine elementos de ambas.

2.1.1 e-Adventure

Esta herramienta ha sido creada por el grupo de investigación <e-UCM> [6] de la Universidad Complutense de Madrid. Es una plataforma para desarrollar juegos de aventuras con fines educativos.

Para usar la herramienta hay que descargarse un archivo zip que contiene el editor para crear los juegos y un motor que se encarga de ejecutarlos.

Al entrar en la herramienta, lo primero que nos encontramos en una pantalla de edición como se puede ver en la Imagen 4. En el menú derecho están todos los elementos necesarios para crear el juego, además hay un menú superior nos encontramos las opciones para crear nuevo juego, guardar, configuración y ayuda de e-Adventure.

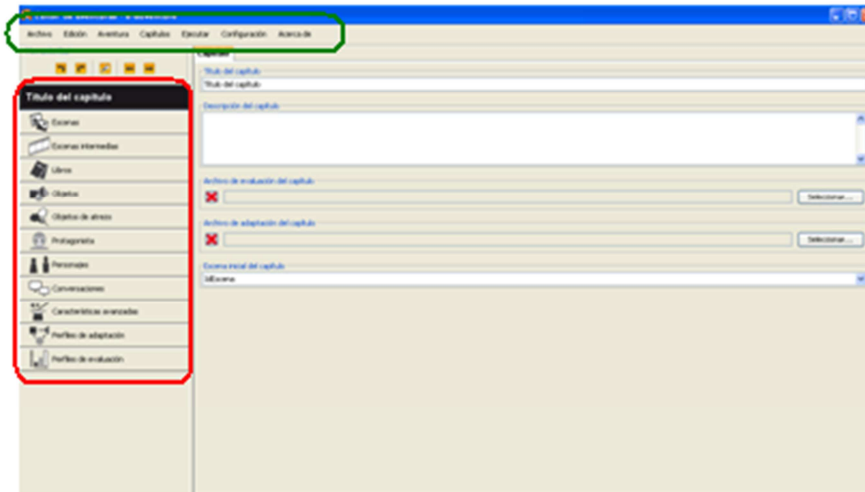


Imagen 4. Pantalla edición e-Adventure Rojo menú derecho, verde menú superior.
Fuente: captura simulador e-Adventure

La estructura de un juego realizado con e-Adventure puede contener los elementos escenas, objetos, personajes, conversaciones y libros.

Para editar cualquier elemento hay que navegar por varias ventanas con formularios para introducir la información necesaria para crear el elemento que queremos.

Los elementos escenas son los lugares donde transcurren las acciones del juego. Para editar una escena las ventanas por las que hay que navegar son, en una se elige la imagen de fondo, en otra se colocan los objetos, otra ventana sirve para introducir información útil para el creador del juego y la última para crear las transiciones entre escenas. Un ejemplo de las distintas ventanas para crear una escena con e-Adventure se puede ver en la Imagen 5.

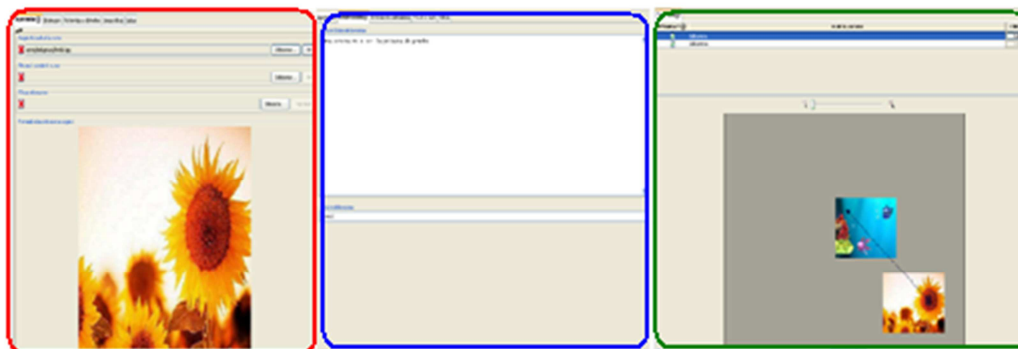


Imagen 5. Crear escena. Rojo ventana imagen fondo, azul información, verde transición escenas.
Fuente: captura simulador e-Adventure

Los elementos objetos pueden ser de atrezzo o para interactuar con el usuario. Para crear un elemento de este tipo también se debe navegar por varias ventanas para introducir la información para crear el juego. La herramienta tiene una ventana de elegir la imagen, otra de guardar información asociada al objeto y la última en la que el usuario puede insertar la acción a realizar. Ejemplo de la creación de un objeto en e-Adventure se muestra en la Imagen 6.

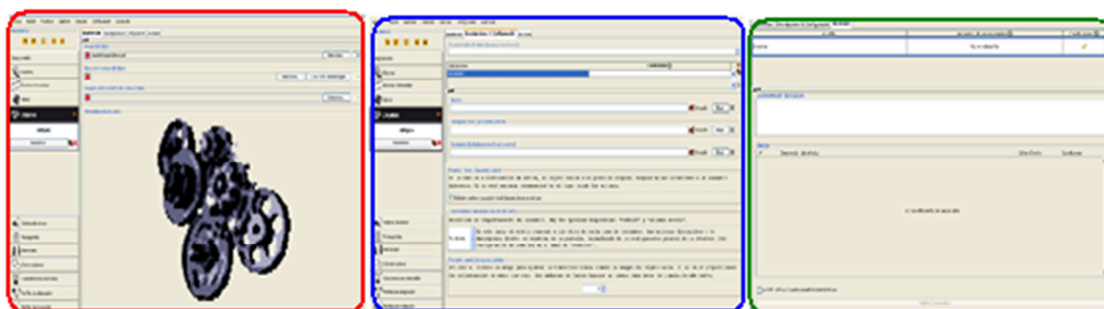


Imagen 6. Crear objeto. Rojo imagen, azul información y verde acciones.
Fuente: captura simulador e-Adventure

Los personajes, son los elementos con los que el jugador interactúa mediante texto durante el desarrollo del juego. En este elemento también se debe de navegar por varias pestañas para crear el personaje, una de las ventanas es para elegir la imagen, otra para el tipo de presentación del texto bocadoillo o cuadro de texto y en la última se edita la acción que quieres realizar con el personaje. Un ejemplo de creación de un personaje con e-Adventure puede verse en la Imagen 7.

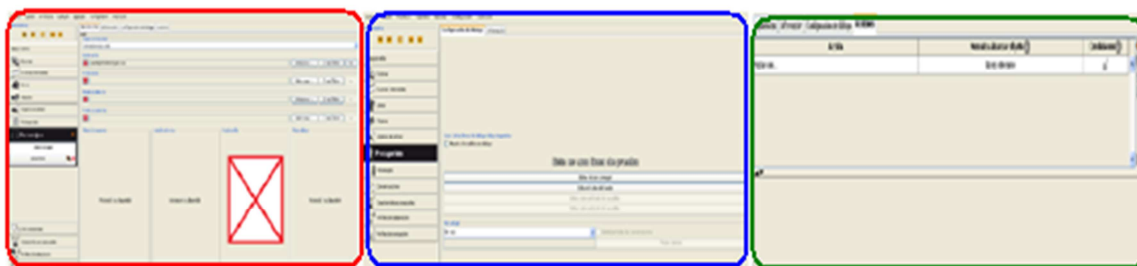


Imagen 7. Crear personaje. Rojo imagen, azul elegir presentación texto y verde acciones con personaje.
Fuente: captura simulador e-Adventure

Los elementos conversaciones son los encargados de crear todo el texto que se introducirá en el juego. Para crear un elemento conversación hay una ventana donde se van añadiendo nodos para entrelazar los textos y los contenidos textuales de esos nodos. En la Imagen 8 se puede ver un ejemplo de un elemento conversación en e-Adventure.

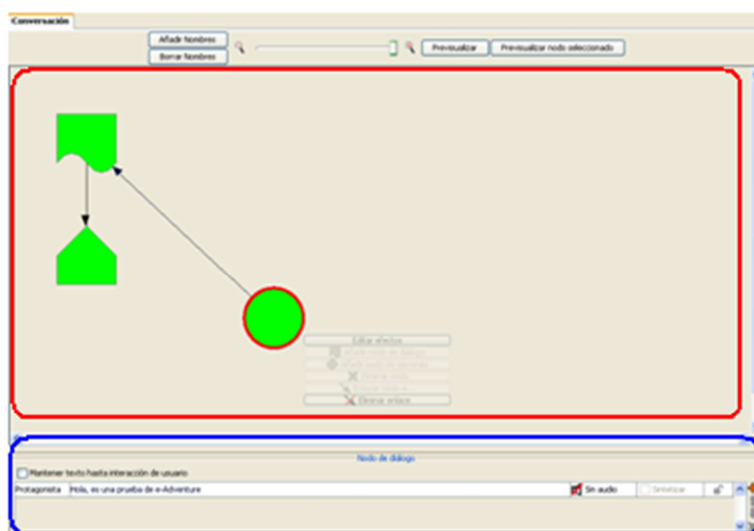


Imagen 8. Crear conversación. Rojo crear nodos y azul introducir textos.
Fuente: captura simulador e-Adventure

Los elementos libros son usados para grandes contenidos de información que el jugador puede consultarlos durante el juego. Para crear este elemento tenemos una ventana para elegir la apariencia y otra para crear el texto que va a contener el libro. Un ejemplo del elemento libro se puede ver en la Imagen 9.

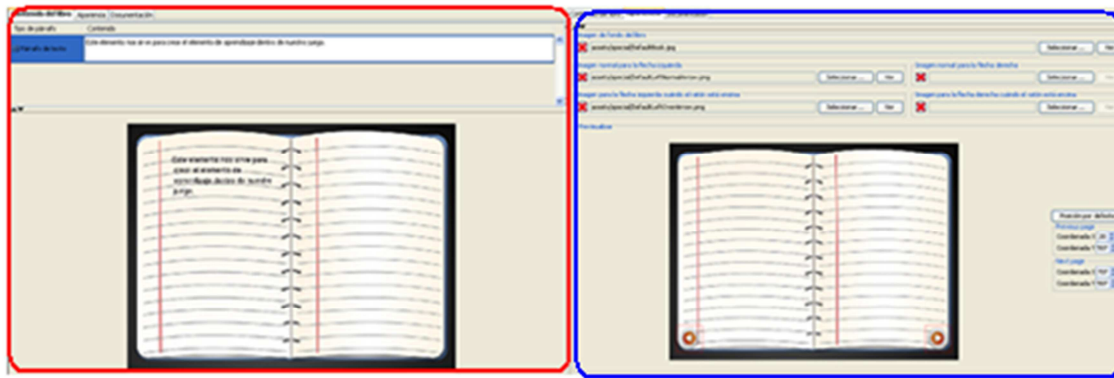


Imagen 9. Crear libro. Rojo introducir texto, azul elegir apariencia.
Fuente: captura simulador e-Adventure

Para el uso de este simulador es necesario consultar un manual [12] para ver qué es cada elemento y cómo se tiene que introducir toda la información necesaria para crearla. Es necesaria una primera ayuda antes de usar esta herramienta porque es bastante compleja de utilizar si es la primera vez que se va a crear un juego con ella. La curva de aprendizaje en esta herramienta es grande.

Tiene una ventaja muy buena que es la gran capacidad de personalización que tiene la herramienta, porque además de poder introducir tus propias imágenes en los elementos escenas, objetos y personajes, tienen más de una opción para crear los textos y la apariencia de los libros. Además los personajes pueden estar animados, es decir, que realicen movimientos por la pantalla.

Otra gran ventaja de los juegos creados con e-Adventure es que cuentan con un seguimiento del jugador en el desarrollo de la aventura.

Esta herramienta en la actualidad tiene un gran uso en educación por su gran capacidad de personalización. El I.E.S Sabón en Arteixo (Galicia) resultó ganador del premio Fundación Telefónica de Innovación Educativa por su juego “El viaje de Alicia con e-adventure: conoce tu ciudad” [6].

Por su alto grado de personalización y su seguimiento del usuario en el juego se ha estudiado esta herramienta para introducir estas características a SimGame.

2.1.2 Storybird

Es una red social de cuentos para crear historias personalizadas con una gran base de datos de imágenes. Al ser una red social, estos cuentos pueden ser compartidos para que mucha gente pueda disfrutar de la historia que crea el usuario.

Para crear una historia lo que hay que hacer es acceder a la página web de Storybird [22] y crearse un usuario según el perfil que prefieras. Hay tres tipos de perfiles como se puede ver en la Imagen 10. El primer perfil es profesor donde el usuario puede

darse de alta a él y a todos sus alumnos con distintas cuentas para que ellos mismo puedan crear su historia siguiendo las indicaciones del educador. El perfil de dibujante permite vender los dibujos propios para que formen parte de las historias que los usuarios crean. Y el último perfil es el de los padres, donde es un usuario normal para crear y compartir el cuento personalizado que cree.

Create, read, and share visual stories.

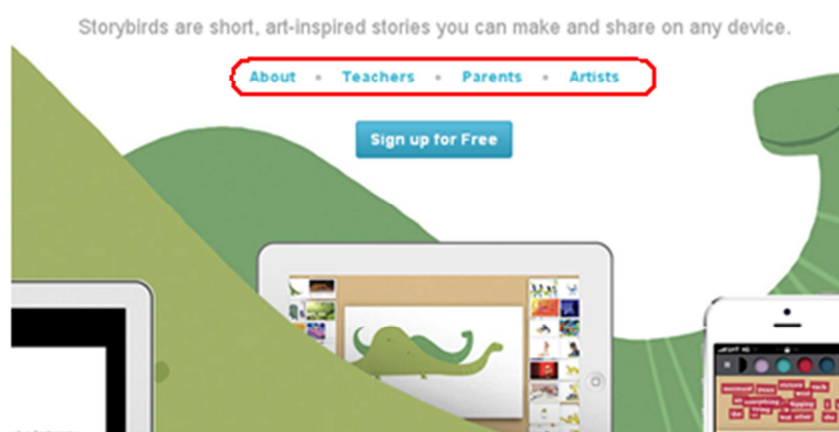


Imagen 10. Pantalla registro Storybird. Rojo perfiles de usuario.

Fuente: captura imagen Storybird

En Storybird hay dos tipos de historias que pueden ser creadas, cuentos o poesías. La historia creada con Storybird puede contener los elementos imagen, texto, hojas y portada.

Para crear una historia lo primero que hay que hacer es elegir la temática del cuento a través de una imagen. Para elegir esta imagen desde la pantalla principal de creación hay un menú lateral derecho con los temas como se puede ver en la Imagen 11.

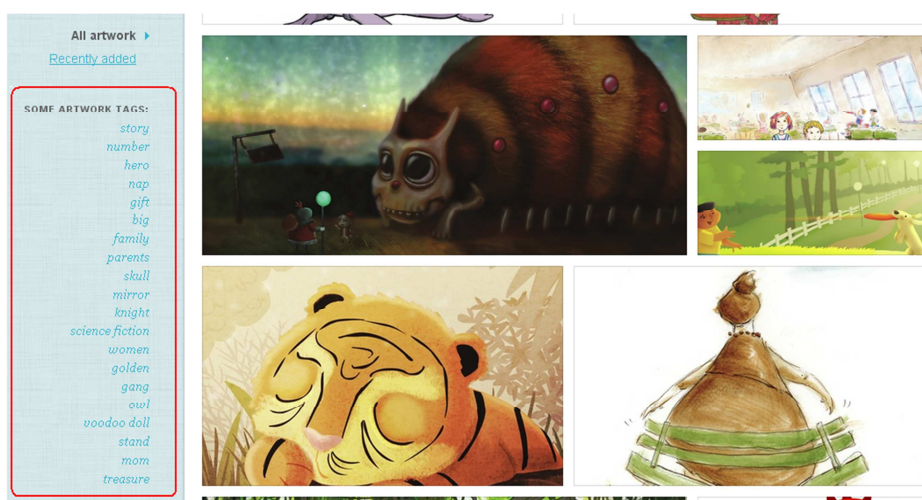


Imagen 11. Pantalla creación Storybird. Rojo menú derecho temas.

Fuente: captura imagen Storybird

El siguiente paso en la creación de la historia es elegir qué tipo de historia quieres crear, hay dos opciones un cuento o una poesía, para ver la pantalla de elección ver la Imagen 12.

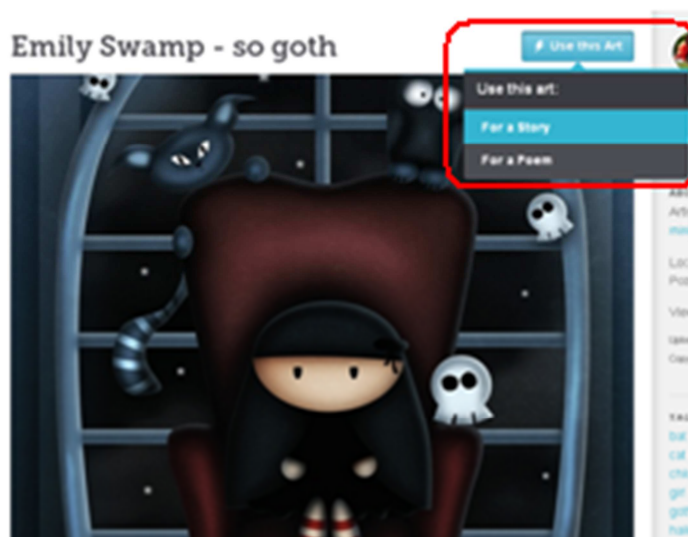


Imagen 12. Pantalla elección historia Storybird. Rojo tipos de historia.
Fuente: captura Storybird

Primero vamos a explicar qué elementos son necesarios para crear un cuento. En la pantalla principal de edición de un cuento hay el cuadro principal donde se colocan la imagen que se quiere editar, una línea inferior donde se van colocando todas las hojas del cuento y en los laterales las imágenes que se pueden incluir en el cuento. Un ejemplo de pantalla principal para crear un cuento se puede ver en la Imagen 13.

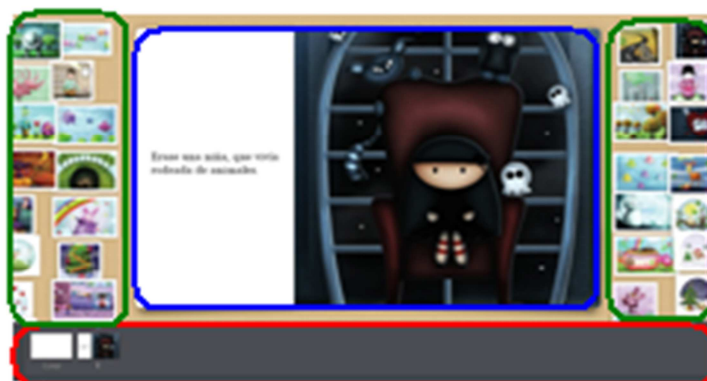


Imagen 13. Pantalla crear cuento. Rojo línea con las hojas del cuento, azul editar hoja y verde imágenes disponibles para el cuento.
Fuente: captura Storybird

Para editar el elemento hoja, añadir una hoja en la línea inferior y seleccionarla para que se sitúe en el cuadro principal.

Para editar el elemento imagen, elegir entre las imágenes que se encuentran en los laterales y arrastrarla hasta situarla dentro de la hoja, se puede elegir entre situar la imagen a la derecha o a la izquierda como se ve en la Imagen 14.



Imagen 14. Editar imagen.
Fuente: captura Storybird

Para editar el elemento texto, situar el cursor en el espacio en blanco para escribir.

Para editar el elemento portada, al entrar a la pantalla de edición del cuento la primera vez se crea automáticamente una hoja en blanco para editar este elemento. Sólo hay que elegir el tipo de portada que queremos y escribir un título. La imagen es la primera que se ha elegido como temática del cuento. Ejemplo de elemento portada se puede ver en la Imagen 15.

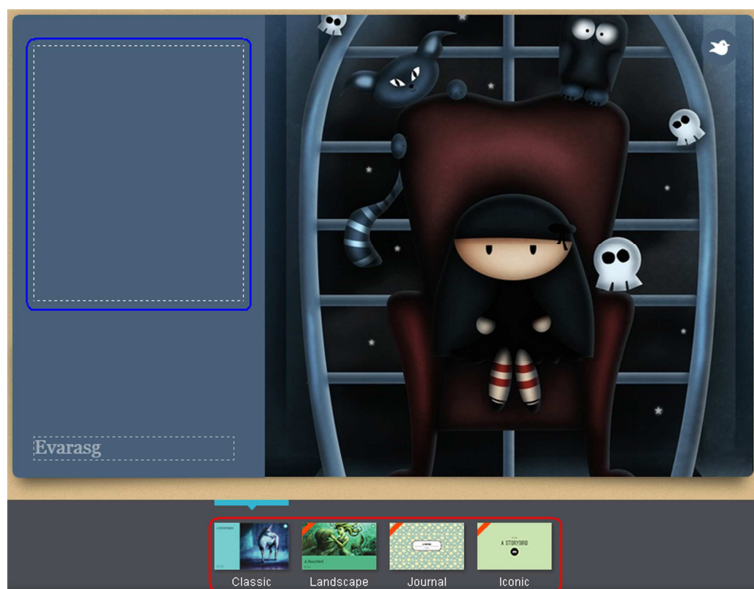


Imagen 15. Editar portada. Rojo tipos portadas y azul editar título.
Fuente: captura Storybird

Para crear una poesía, tenemos los elementos imagen. En la pantalla principal de edición de una poesía tenemos la imagen elegida del tema en un marco de edición, unos colores para los textos y una bolsa de palabras para crear el poema, como se puede ver en la Imagen 16.

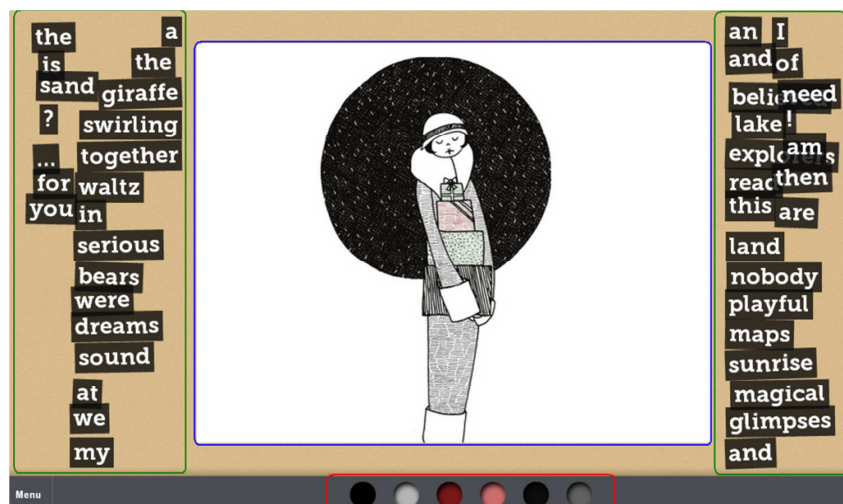


Imagen 16. Pantalla edición poema. Rojo colores texto, azul cuadro editar poema y verde textos.
Fuente: captura Storybird

Para editar el elemento texto, hay que arrastrar las palabras dentro de la imagen en el cuadro principal de edición y situarla en el lugar que se quiera. Ejemplo editar texto en Imagen 17.

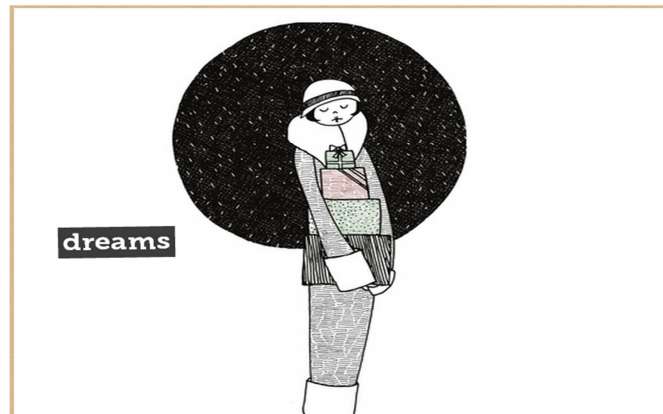


Imagen 17. Editar texto.
Fuente: captura Storybird

El uso de Storybird es bastante intuitivo y guiado durante todo el proceso de creación, crear un cuento o un poema no supone mucho tiempo en edición.

Aunque esta herramienta dispone de una amplia base de datos de imágenes, la personalización de la historia es mínima, ya que sólo se puede elegir la temática de la historia eligiendo una imagen y las posiciones de las imágenes y los textos dentro de las hojas.

Tampoco tiene ningún tipo de seguimiento de usuario cuando estás viendo un cuento, si el usuario desea ver un cuento que han publicado por alguna razón sale de la página, cuando vuelve al cuento debe de empezar desde el principio.

Por su sencillez esta herramienta ha sido estudiada para diseñar el proceso de guiado del usuario dentro de SimGame de modo que disminuya la curva de aprendizaje de la herramienta.

2.1.3 Conclusiones del Estado del Arte

Tras analizar el estado del arte se ha decidido crear una herramienta que incorpore parte de las ventajas de herramientas de autoría de juegos como e-Adventure como su flexibilidad para personalizar los elementos del juego, colocarlos en el escenario, personalizar los textos y la inclusión en el juego de vídeos y parte de la facilidad de uso de la herramienta de autoría de narrativa, al crear una interfaz intuitiva y vistosa.

Finalmente para crear SimGame se han seleccionado los siguientes elementos de juego: terrenos, objetos, retos, tesoros, formularios, textos, personajes, animaciones y vídeos. Además de crear un flujo de simulación guiado para facilitar la edición a personas sin conocimientos técnicos.

Tras barajar las diferentes opciones de diseño finalmente se ha optado por introducir la información de todos los elementos del juego (terrenos, objetos y tesoros) en un

único formulario, para facilitar aún más la edición algunas de las entradas son menús desplegables y casillas de elección o checkbox.

Los datos asociados a los retos del juego se presentan en dos formularios consecutivos, el primero sirve para crear la presentación del reto y el segundo para crear el mini-juego. Cuando el usuario termina de introducir los datos en el primer formulario y pulsa guardar, el segundo formulario es mostrado.

Los elementos de ayuda asociados a tesoros, objetos, retos, terrenos, flujo y crear ficheros están en componentes gráficos de tipo ActioPoint uno por cada elemento para facilitar la navegación por el simulador. Así al agrupar cada acción en un ActionPoint la herramienta se simplifica.

Al comienzo de SimGame se le muestra al usuario una ayuda del simulador bastante sencilla, ya que el uso del simulador es sencillo e intuitivo.

Para crear un juego, lo primero que se debe de saber es la estructura y los elementos que van a aparecer, esta es la fase más complicada de crear el juego ya que la fase de edición es rápida.

El juego resultante es bastante personalizado, ya que todas las imágenes, textos, vídeos y elementos de los juegos los introduce el usuario.

Tanto SimGame como el juego o simulador que se obtiene como resultado de usar la herramienta tienen un seguimiento de usuario que permite entre otras cosas guardar el estado del juego, de modo que cuando se cierre el juego y se vuelva a entrar, al usuario aparece en el mismo punto del juego en el que se encontraba cuando salió del sistema.

La Tabla 3 resume las principales características de SimGame en relación con las dos herramientas analizadas.

Tabla 3. Conclusiones Estado del Arte

	e-Adventure	Storybird	SimGame
Capacidad de personalización	Alta	Baja	Alta
	Escenas	Imágenes	Terrenos
	Objetos	Hojas	Objetos
	Personajes	Textos	Retos
	Conversaciones	Portada	Tesoros
Elementos juego	Libros		Textos
	Textos		Personajes
	Vídeos		Formularios
			Vídeos
			Mini-juegos
Registro progreso usuario	Si	No	Si

Curva de aprendizaje	Alta	Baja	Media
Tiempo necesario creación	Grande	Corto	Medio
Proceso de edición guiado paso a paso por la herramienta	No	Si	Si
Editar juego una vez creado	Si	Si	Si
Perfil usuario crear juego	Perfil técnico	Cualquier usuario	Cualquier usuario

2.2 Estándares empleados

Los estándares que se han empleado para desarrollar SimGame son.

2.2.1 XML

Es un lenguaje de programación desarrollado por el Word Wide Web Consortium (W3C) [28].

Este lenguaje proporciona una gran flexibilidad ya que se pueden definir nuestras propias etiquetas según las necesidades de nuestro proyecto. Esto también nos facilita la interpretación de los mismos.

En SimGame hemos usado este estándar para crear el fichero de configuración y los ficheros de los escenarios, estos ficheros definen los recursos que van a formar parte de nuestro simulador (ver anexo 1).

También son XML los ficheros de los escenarios del juego que crea el educador y el fichero de configuración.

2.2.2 SCXML

Este estándar también ha sido desarrollado por el Word Wide Web Consortium (W3C). [28]. Es un lenguaje basado en XML para definir máquinas de estados.

En la página oficial de W3C [28] explica esta tecnología como una estructura de máquina de estados donde los conceptos básicos son los estados, las transiciones y los eventos. Cada estado tiene un conjunto de transiciones que definen cómo reacciona a los eventos. Los eventos pueden ser generados por instrucciones en el mismo fichero o por instrucciones externas al fichero.

Esta tecnología ha sido utilizada dentro de SimGame para definir el flujo dentro del simulador y también para definir el flujo dentro del juego del educador (ver anexo 2).

2.2.3 HTML

Es el lenguaje utilizado para la creación de páginas web, fue creado a partir de SCXML, y como en los estándares anteriormente mencionados también está a cargo del W3C [28].

Con la evolución de Internet que en principio fue creado para mandar información de texto, en la actualidad Internet es más utilizado para el envío de información multimedia en diferentes dispositivos. Por esta razón este estándar ha ido evolucionando en diferentes versiones, siendo la última HTML5 el cual permite diseños adaptables para distintos dispositivos.

En SimGame se utiliza HTML para todas las ventanas de formularios (ver anexo 3) y pop ups que muestran la información. También es utilizado en el juego creado por el educador por si quiere introducir formularios, en la presentación de los retos para meter vídeo y texto y en los juegos de clasificar y ordenar para crear la estructura del juego.

2.2.4 JavaScript

Es un lenguaje de programación interpretado, es decir, no requiere compilación. Es soportado por la mayoría de los navegadores como Internet Explorer, Mozilla Firefox y Opera entre los más utilizados.

JavaScript nos permite crear una página dinámica que pueda interactuar con el usuario.

En SimGame se utiliza JavaScript tanto en los elementos gráficos que se presentarán al usuario final (ej: juegos de clasificar y ordenar) como para tareas de gestión de la herramienta de edición como por ejemplo, extraer o guardar información en la base de datos, crear formularios dinámicos o modificar algún elemento del juego.

2.2.5 MySQL

Es un sistema de administración de base de datos que utiliza múltiples tablas para almacenar y guardar la información.

Permite interactuar con lenguajes como JavaScript o php y por eso es uno de los más utilizados, además de ser open source.

En SimGame se ha utilizado tanto en la fase de diseño para guardar toda la información que introduce el educador para definir los elementos del juego, como en la fase de despliegue para guardar la información de interacción del usuario final con el juego creado.

2.3 Herramientas necesarias

En este apartado se van a presentar las herramientas necesarias para que educador y alumno puedan ejecutar sin ningún tipo de problema el simulador en el ordenador.

2.3.1 Simuflow

Esta herramienta [27] desarrollada en un proyecto fin de carrera anterior proporciona una biblioteca básica de componentes gráficos personalizables y reutilizables desarrollados en flash y JavaScript que se utilizarán para presentar la información al usuario final, así como una arquitectura de referencia para su despliegue y utilización.

El trabajo presentado extiende Simuflow de varias formas (1) Creando componentes de alto nivel que representan elementos del juego (escenarios, tesoros, retos, objetos) a partir de los elementos gráficos básicos basados en flash ver Imagen 18. (2) Creando un editor que permite la creación y personalización de estos elementos a partir de formularios. (3) Modificando los juegos básicos creados en Simuflow de clasificación y ordenación para que puedan ser editados según las necesidades del educador. (4) Elaborando un simulador que guía al usuario en todo proceso hasta elaborar un simulador plenamente funcional.

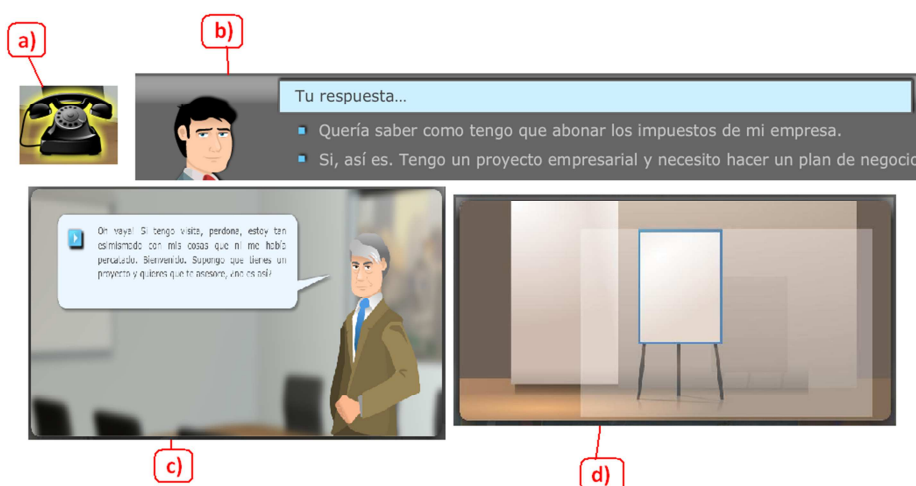


Imagen 18. Componentes gráficos de Simuflow: (a) ActionPoint, (b) UserActionPanel, (c) DialoguePopUp, (d) Animation

Fuente: Simuflow <http://e-archivo.uc3m.es/handle/10016/16055>

2.3.2 XAMPP

El simulador guarda toda la información en una base de datos MySQL, para tener esta base de datos necesitamos instalar un servidor web Apache. Para personas no técnicas, que son los principales clientes de nuestro simulador, es difícil de instalar un servidor en su ordenador, lo que nos proporciona esta herramienta es la facilidad a la hora de la instalación ya que contiene PHP y MySQL que son necesarios.

Otra de las razones de su elección, es que es un recurso gratuito lo que hace que los costes disminuyan.

Para descargar XAMPP hay que acceder a la página web de Apachefriends [2], ir a la sección de descargas y elegir el sistema operativo. De los tres modos de instalación de

los que disponemos, se recomienda elegir la opción de “Instaler” ya que es la opción de instalación más sencilla.

Cuando tengamos el ejecutable en nuestro ordenador, damos doble click y sólo es seguir las instrucciones ver Imagen 19.

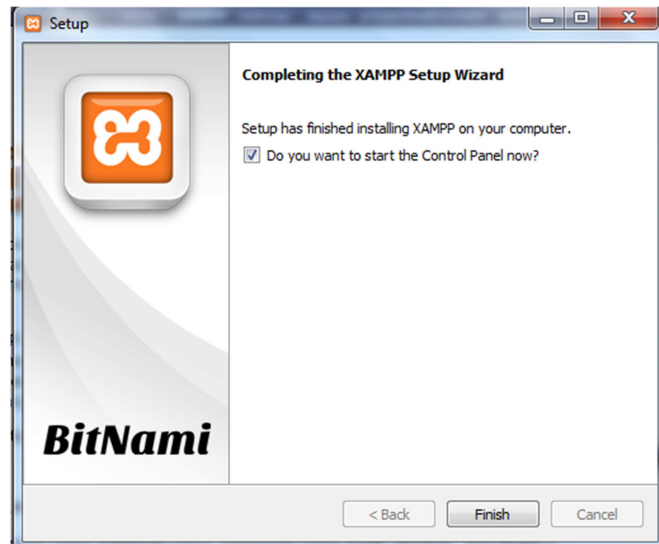


Imagen 19. Pantalla instalación XAMPP.
Fuente: captura instalación XAMPP

Cuando el usuario tenga que arrancar el XAMPP lo único que tiene que hacer es ir a su bandeja de inicio o en el icono que se suele poner en el escritorio, pincharle para que se abra y dar al botón de “Star” en Apache y MySQL. En la Imagen 20 se puede ver el panel de control de XAMPP.

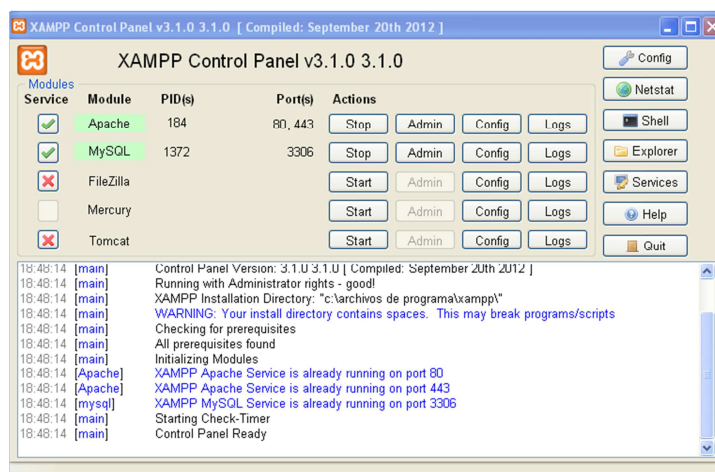


Imagen 20. Panel de control XAMPP.
Fuente: captura instalación XAMPP

Para acceder al panel de control de la base de datos tendrá que introducir en el navegador la dirección 127.0.0.1 y a través del panel de la Imagen 21 acceder a phpMyAdmin.



Imagen 21. Web 127.0.0.1. Fuente: 127.0.0.1

2.3.3 Adobe Flash Player

Flash Player es una herramienta de Adobe que permite reproducir películas Flash. En Internet un gran número de juegos que están disponibles online usan esta plataforma.

Es soportado por todos los navegadores ya que en muchas páginas web es necesario tenerlo instalado para un correcto funcionamiento y visualización de la página. Aunque con el lanzamiento de HTML5 el uso de Flash ha disminuido.

Cualquier usuario puede descargarse esta herramienta gratuitamente desde la página de Adobe [1].

2.3.4 Navegador Firefox Mozilla

Es un navegador libre, es decir, el usuario puede descargárselo gratis desde la página de Mozilla [16], desarrollado para los sistemas operativos Windows, Mac y Linux.

El simulador ha sido optimizado para este navegador, aunque en Chrome funciona presenta el inconveniente de que pueden ocurrir comportamientos no deseados o no presentarse correctamente el estilo de la interfaz gráfica. Lo que se aconseja instarse el navegador Firefox Mozilla.

Capítulo 3

Diseño y Arquitectura del Sistema

En este apartado se analizan las decisiones de diseño adoptadas y se describe en detalle la arquitectura del simulador.

3.1 Descripción general

SimGame consiste en una herramienta de autor que agiliza el proceso de creación de simuladores educativos guiando al usuario en la creación de escenarios, objetos, tesoros y diferentes situaciones o retos que tendrá que resolver finalmente el alumno. Para la creación de los simuladores los autores seleccionan un conjunto de elementos multimedia como imágenes, diálogos y vídeos que ilustren a los personajes, y los elementos interactivos que ayudan a la organización del juego basada en la estructura narrativa de presentación-nudo-desenlace. El proceso de guiado de SimGame a través de la funcionalidad de la herramienta también sigue una estructura similar presentación-nudo-desenlace donde primero se da una visión global del simulador, a continuación se pide los datos necesarios para crear el juego y por último se crean los ficheros personalizados necesarios que guardan la información que permite desplegar el simulador en otros equipos.

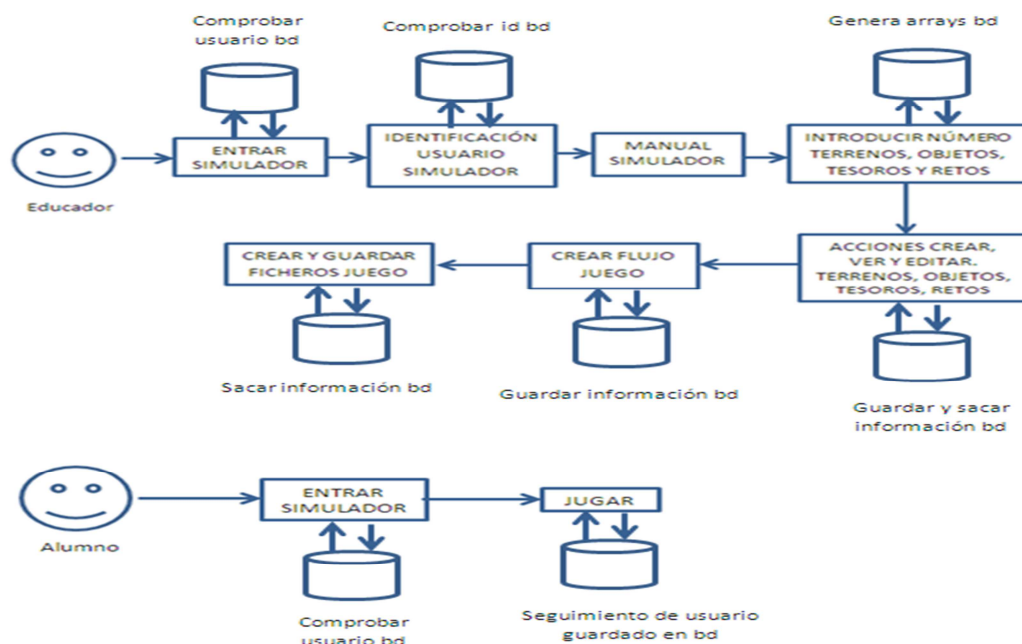


Imagen 22. Diagrama casos de uso SimGame. En el sistema actual hay una única base de datos para almacenar información de los educadores y una base de datos por cada nuevo simulador creado por el usuario final, para almacenar la información de sus alumnos. Estas bases de datos se pueden distribuir para almacenar por separado diferentes tipos de información.

3.2 Requisitos funcionales y técnicos

En este apartado se van a describir los requisitos funcionales y técnicos para el desarrollo de SimGame.

3.2.1 Requisitos funcionales

Tabla 4. Requisitos funcionales simulador SimGame

Identificador	Requisito	Descripción
RF - 1	Personalizable	Tiene que crear juegos y simuladores que cumplan las necesidades específicas del usuario, permitiéndole incluir sus propios recursos multimedia.
RF - 2	Sencillez	SimGame debe de ser una herramienta sencilla para que cualquier perfil de usuario pueda utilizarla siguiendo un procedimiento guiado.
RF - 3	Rapidez en la creación del juego	SimGame es una herramienta bastante guiada y diseñada según el requisito RF-2 para que crear un juego no suponga emplear mucho tiempo.
RF - 4	Sin conocimientos de programación	Cualquier usuario debe de poder crear un juego con SimGame. Pueden ser usuarios más técnicos donde si sepan programar o perfiles no técnicos pero que necesiten crear un juego personalizado.
RF - 5	Sin conocimientos en edición de imágenes	Para crear un juego con SimGame no hace falta usar las herramientas de edición de imágenes, si no que cualquier imagen que tenga en su ordenador puede ser introducida en el simulador

3.2.2 Requisitos técnicos

Tabla 5. Requisitos técnicos del simulador SimGame

Identificador	Requisito	Descripción
RT - 1	Componentes: pantallas, imágenes, mini-juegos, textos, personajes, vídeos y formularios	En el juego del usuario se pueden incluir todos estos componentes totalmente personalizables con los recursos con los que disponga.
RT - 2	Guardar información en arrays BD	Para guardar toda la información que el usuario introduzca en SimGame para crear su juego, debe de estar organizado en arrays dentro de la base de datos
RT - 3	Crear ficheros personalizados en ventanas separadas	Para facilitar la fase de guardar los archivos personalizados, es necesario que los ficheros se creen en una ventana diferente a la del

		simulador.
RT - 4	Registro progreso de usuario	SimGame debe de matener un registro del progreso del usuario, para cuando el jugador abandone el juego personalizado al volver a entrar esté situado en la pantalla donde los dejó.
RT - 5	Sistematizar simuladores	SimGame debe seguir unos patrones básicos en la creación de simuladores para garantizar una coherencia en la estructura narrativa y en los elementos gráficos utilizados para presentar la información e interactuar con el usuario.

3.3 Diseño. Elementos gráficos

En este apartado se describen los diferentes elementos gráficos utilizados para la presentación de la información en pantalla. Algunos de estos elementos han sido heredados de SimufLOW y otros han sido modificados o creados desde cero para SimGame.

3.3.1 Elementos gráficos de SimGame

SimGame utiliza algunos de los elementos gráficos de las librerías de SimufLOW representados en la Imagen 18 para presentar la información al usuario y permitir seleccionar diferentes caminos en la narrativa. En SimGame se han creado formularios que permiten crear y personalizar estos elementos desde un formulario web sin necesidad de tener conocimientos de HTML, Flash o Javascript:

- **ActionPoint:** Es un componente flash que representa un elemento seleccionable representado por una imagen rodeada de un halo de color amarillo. Tanto la imagen como su ubicación en pantalla pueden ser modificadas por el usuario.
- **Dialogue PopUp:** Es un componente flash que contiene un componente de diálogo y un personaje. Tanto el texto, el fondo del Dialogue PopUp como el personaje se pueden personalizar.
- **UserActionPanel:** Es un componente flash que consiste en una imagen, una pregunta y tres opciones de respuesta. Tanto la imagen como el contenido de la pregunta y las respuestas es personalizable. Las tres respuestas pueden ser etiquetadas como correcta, fallo leve y fallo grave.
- **Imagen, Vídeo, Animation y PopUpWindow.** Son contenedores genéricos que permiten almacenar una imagen, un vídeo, una animación o una combinación de texto e imagen para presentar la información. Las animaciones y los vídeos se suelen utilizar para dar un aspecto más dinámico al simulador representando acciones comunes (abrir una puerta, enviar un correo, coger un teléfono o para transiciones entre escenas). En cualquiera de estas formas de presentar la

información el contenido se puede personalizar mediante recursos externos seleccionados por el usuario.

Además de estos elementos básicos contiene otros elementos más complejos como (1) juegos o (2) formularios HTML que se han modificado durante la elaboración de este proyecto y que permiten recoger interacción del usuario e información relevante de su desempeño en el juego. Al igual que ocurría con los componentes anteriores en SimGame además de modificar estos elementos ha creado formularios que permiten crear estos componentes a partir de un sencillo formulario web.

3.3.2 Juego de clasificar

Este elemento es un componente HTML, al alumno se le presenta un juego donde tiene unos elementos que debe de situar en la categoría a la que pertenezcan.

El educador en este elemento puede personalizar el número de elementos, el contenido de cada elemento que puede ser texto o imagen, la correspondencia entre elemento/categoría y el nombre de las categorías. Para que el proceso de correspondencia entre elemento y categoría sea más sencillo, se le presenta un menú desplegable por cada elemento con las categorías que hay, ver Imagen 23.

Imagen 23. Formulario juego clasificar.
Fuente: captura SimGame

3.3.3 Juego de ordenar

Este elemento es un componente HTML, donde se le presenta al alumno un juego donde tiene una columna de elementos que debe de arrastrar y colocar en un orden.

El educador en este elemento puede personalizar el número de elementos, el contenido de cada elemento que puede ser texto o imagen y el orden de cada elemento en el juego. Al introducir las posiciones de cada elemento en el juego se presenta al educador un menú desplegable con todas las posiciones como se puede ver en la Imagen 24 para que este proceso de creación sea más sencillo.

Imagen 24. Formulario juego ordenar.
Fuente: captura SimGame

3.2.3 Ventanas de formulario

Este elemento es un componente HTML, donde se le presenta al alumno un formulario con unos campos que tiene que rellenar. Se han utilizado elementos de este tipo para facilitar al usuario final la creación de todos los elementos necesarios para el despliegue del simulador, tanto los elementos gráficos como los elementos de interacción o el flujo de simulación que decide el orden de presentación de todos estos elementos en pantalla. También se permite al profesor usuario de SimGame crear sus propios formularios que serán presentados durante la simulación a sus alumnos y le permitirán recoger información durante el proceso de simulación (proporcionar sus datos, responder preguntas, enviar dudas, etc.). A continuación se describe el proceso que debería seguir un usuario de SimGame para crear formularios para sus alumnos.

El educador en este elemento puede personalizar el número de entradas en el formulario, el tipo de cada entrada donde puede elegir entre campo de texto, radio buttons y menú desplegable y también puede personalizar el número de elementos que tiene cada entrada ver Imagen 25. Para introducir varias entradas en el formulario, lo que tiene que hacer es cuando termina de rellenar los campos del formulario pulsar el botón “Siguiente input”, cuando no desee introducir más entradas con pulsar “Guardar y terminar” ya tendrá todo el formulario guardado en la base de datos.

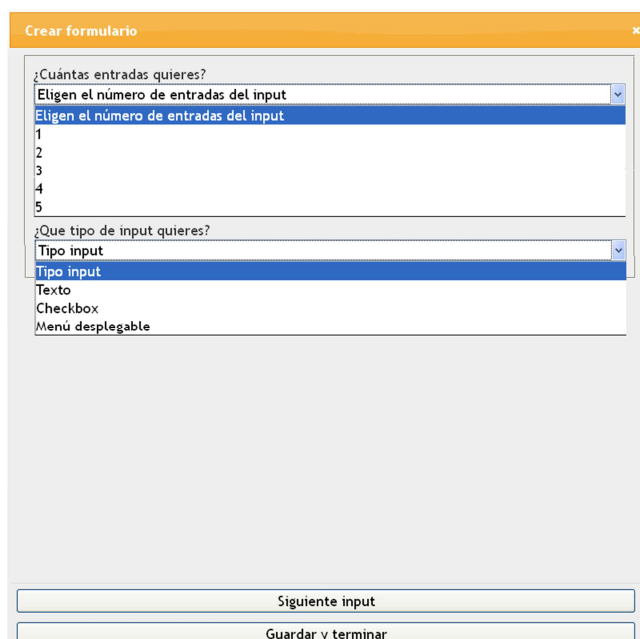


Imagen 25. Formulario crear formulario con número inputs variable.
Fuente: captura SimGame

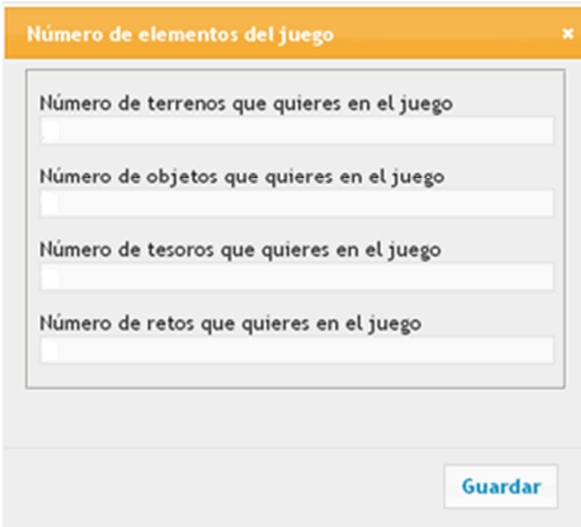
En SimGame se han creado ventanas de formulario para (1) introducir el número de elementos para los terrenos, objetos, tesoros y retos (2) crear terrenos, objetos, tesoros y retos, (3) editar terrenos, objetos, tesoros y retos, (4) crear formularios, (5) crear el flujo del juego del educador, (6) crear los ficheros del juego del educador y (7) menús para ver y editar los elementos.

3.4 Diseño. Elementos del simulador

En esta sección se describen los elementos del simulador que son los componentes principales del juego que se va a crear. Estos elementos se presentan al usuario utilizando uno o varios de los elementos gráficos presentados en el apartado anterior.

Nada más entrar al simulador se ofrece al usuario la posibilidad de ver el proceso de creación de un simulador mostrando una guía con cada elemento por separado. A

continuación debe de introducir el número de elementos como se ve en el formulario de la Imagen 26.



Número de elementos del juego

Número de terrenos que quieres en el juego

Número de objetos que quieres en el juego

Número de tesoros que quieres en el juego

Número de retos que quieres en el juego

Guardar

Imagen 26. Formulario para introducir el número de elementos en el juego.
Fuente: captura SimGame

Cuando ya se ha rellenado este formulario se pasa a la pantalla principal de SimGame, donde se utiliza un contenedor o imagen de fondo con un conjunto de ActionPoints que representan a cada uno de los elementos del juego: terrenos, objetos, tesoros, retos, flujo, formularios, ficheros, ayuda y probar mini-juegos Imagen 27 i).

Tanto la imagen de fondo como los elementos seleccionables (aspecto y ubicación) se pueden personalizar. Se soportan diferentes formatos de imagen como (.png, .jpg, .gif, .swf).

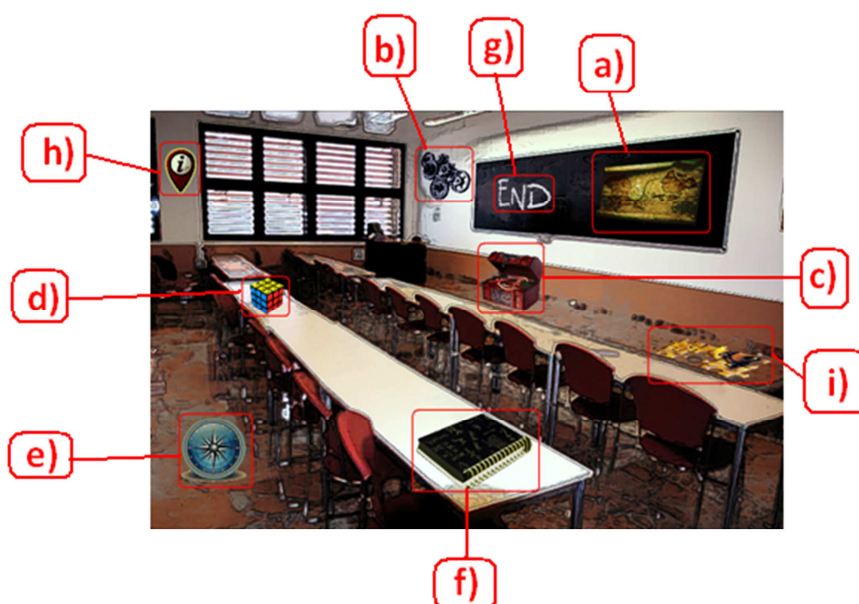


Imagen 27. Pantalla principal que da acceso a todos los elementos del juego a)terrenos, b)objetos, c)tesoros, d)retos, e)flujo, f)formularios, g)crear ficheros, h)ayuda, i)Probar mini-juegos.

Fuente: captura SimGame

Al pulsar sobre el elemento h) de la Imagen 27 se muestra una guía que informa al usuario sobre la utilidad de cada elemento para el diseño del juego. Son ventanas escritas en HTML y las acciones que se realizan en estas ventanas son, pulsar los botones que aparecen para continuar con la explicación o cerrar la ventana.

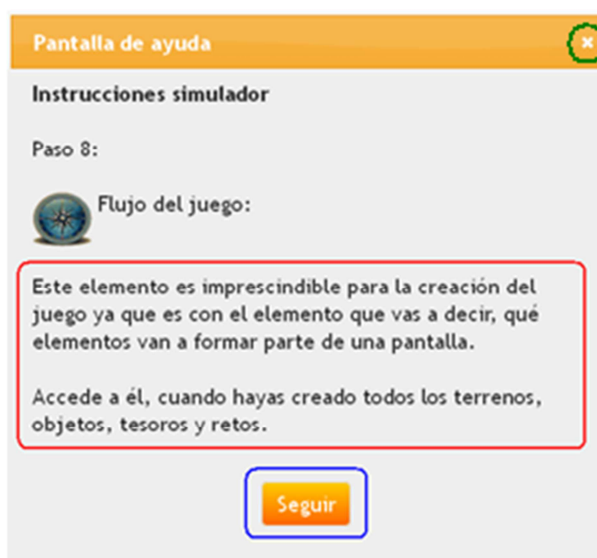


Imagen 28. Pantalla ayuda simulador. Rojo explicación elemento, azul botón continuar ayuda y verde cerrar ayuda.

Fuente: captura SimGame

A continuación se muestran los elementos del simulador.

Tabla 6. Descripción elemento terrenos del simulador SimGame


Nombre	Terrenos
Imagen	
Descripción	<p>Este elemento es el equivalente a las pantallas de un juego representan la imagen de fondo de los diferentes escenarios. El educador debe de introducir:</p> <ul style="list-style-type: none"> -Número de terreno, elegir una opción de un menú donde se muestran el número de terrenos que ha introducido. -Nombre, debe de estar formado por una palabra ya que luego se nombrarán los archivos del juego con este nombre. -Descripción, le sirve de información al educador. -Nombre de la imagen que desea situar de fondo.

Tabla 7. Descripción elemento objetos del simulador SimGame


Nombre	Objetos
Imagen	
Descripción	<p>Estos elementos pueden ser objetos decorativos en la interfaz representados mediante imágenes superpuestos a la imagen principal o elementos seleccionables (ActionPoints) que permitan disparar una acción como cambios en la interfaz, en la BD o paso de un escenario a otro. El educador debe de introducir:</p> <ul style="list-style-type: none"> -Número de objeto, elegir una opción de un menú donde se muestran el número de objetos que ha introducido. -Nombre. -Descripción, le sirve de información al educador. -Nombre de la imagen a mostrar en el juego. -Posición, para situar la imagen en la pantalla del juego. -Elegir si va a ser un objeto de transición entre terrenos.

Tabla 8. Descripción elemento tesoros del simulador SimGame



Nombre	Tesoros
Imagen	
Descripción	<p>Estos elementos sirven de recompensa al superar los retos. Además de poder disparar acciones como los objetos pueden actualizar la información de progreso del usuario en la base de datos. El educador debe de introducir:</p> <ul style="list-style-type: none"> -Número de tesoro, elegir una opción de un menú donde se muestran el número de tesoros que ha introducido. -Nombre. -Descripción, le sirve de información al educador. -Nombre de la imagen a mostrar en el juego. -Posición, para situar la imagen en la pantalla del juego.

Tabla 9. Descripción elemento retos del simulador SimGame

Nombre	Retos
Imagen	
Descripción	<p>Estos elementos sirven para crear juegos. Se componen de dos partes, primero ha de crearse la presentación una explicación que sirva de introducción al problema que hay que resolver. Para realizar esta presentación pueden utilizarse contenedores genéricos (Image, Video, Animation, PopUpWindow) o elementos de diálogo con un personaje (DialoguePopUp) Y la otra parte es la creación del reto, que puede consistir en realizar un pequeño juego de clasificación u ordenación (Game) o resolver alguna pregunta (UserActionPanel). El educador debe de introducir:</p> <ul style="list-style-type: none"> -Número de reto, elegir una opción de un menú donde se muestran el número de retos que ha introducido. -Nombre, debe de estar formado por una palabra ya que se nombrarán con este nombre los archivos de los juegos. -Descripción, le sirve de información al educador. -Nombre de la imagen a mostrar en el juego. -Posición, para situar la imagen en la pantalla del juego. -Tipo de presentación: texto, vídeo, Animation o

DialoguePopUp.

-Información necesaria para crear el elemento del tipo de presentación elegido.

-Datos para crear los juegos o el UserActionPanel, dependiendo de lo que se elija.

Tabla 10. Descripción elemento formularios del simulador SimGame


Nombre	Formularios
Imagen	
Descripción	<p>Este elemento permite crear formularios que pueden tener tantas entradas como desee el educador y pueden ser de tipo, texto, checkbox y menú desplegable (Form). La información introducida por el usuario en estos formularios se almacena en la base de datos y puede utilizarse para decidir el flujo que seguirá la simulación. El educador debe de introducir:</p> <ul style="list-style-type: none"> -Nombre del formulario, debe de estar formado por una palabra ya que se nombrarán los archivos de los formularios con este campo. -Nombre de la imagen a mostrar en el juego. -Posición, para situar la imagen en la pantalla del juego. -Tipo de las entradas en el formulario. -Número de elementos en cada entrada.

Tabla 11. Descripción elemento probar mini-juegos del simulador SimGame


Nombre	Probar mini-juegos
Imagen	
Descripción	<p>En este elemento permite ver y jugar a los juegos de clasificar y ordenar que han sido creados para los retos. El educador lo único que debe de introducir elegir en un menú el juego que quiere probar. Hay dos opciones donde se separan los juegos de clasificar de los juegos de ordenar.</p>

Tabla 12. Descripción elemento ayuda del simulador SimGame


Nombre	Ayuda
Imagen	
Descripción	<p>En este elemento contiene dos guías.</p> <ol style="list-style-type: none"> 1. Una guía con los elementos que forman el simulador y una pequeña explicación de para qué sirve cada uno. 2. Una guía con “La aventura del héroe” que es una estructura narrativa en 12 pasos basado en el esqueleto tradicional presentación-nudo-desenlace.

Tabla 13. Descripción elemento flujo del simulador SimGame



Nombre	Flujo
Imagen	
Descripción	<p>Este elemento permite orquestar el flujo de la simulación. Para hacerlo el usuario establece el orden en el que se despliegan los diferentes elementos del simulador y qué condiciones se tienen que dar (nivel de desempeño, eventos de usuario, etc.) para que se despliegue cada parte de la trama argumental.</p> <p>El educador debe de elegir que terreno quiere crear el flujo y de una lista elegir los objetos, retos y formularios creados para que aparezcan en cada pantalla del juego</p>

Tabla 14. Descripción del elemento crear ficheros del juego del simulador SimGame

Nombre	Crear fichero juego
Imagen	
Descripción	<p>En este elemento se crean los ficheros del juego. El educador debe de:</p> <ul style="list-style-type: none"> -Elegir los terrenos para crear los ficheros asociados a él. -Elegir la opción de crear el fichero de configuración donde debe de introducir el nombre de la base de datos que ha creado y elegir cuál será la primera pantalla del juego.

3.5 Diseño. Arquitectura

El sistema se divide en varios módulos que se describen a continuación y pueden verse en la Imagen 29 .

El módulo core es el encargado de:

- Gestionar el simulador. Paquete SimuFlowManager.
- Gestión de los datos en ficheros y en la base de datos. Paquete DataHandler.
- Definir los elementos de la interfaz. Paquete Model.
- Gestionar la interfaz gráfica. Paquete InterfaceManager.
- Crear el seguimiento del usuario. Paquete Tracking.
- Crear funcionalidad adicional a la gestión propia de los eventos Flash. Paquete EventManager.

El módulo HTML contiene:

- Paquete forms, formularios y pantallas del simulador SimGame que se presentan al educador para crear el juego. Están compuesto por clases que crean:
 - Menús dinámicos que sirven para elegir el elemento a ver o editar.
 - Formularios para introducir los datos de los terrenos, objetos, tesoros, retos y formularios.
 - Formulario inicial para identificar al usuario.
 - Formulario para introducir el número de elementos de terrenos, objetos, tesoros y retos.
 - Formulario para crear el flujo del juego.
 - Formulario para crear los ficheros personalizados del juego del educador.
- Paquete edit, formularios con los mismos campos que los formularios del paquete forms a excepción del campo nombre que no puede ser modificado. Hay formularios para editar terrenos, retos, objetos, tesoros y formularios.
- Paquete saveas, pantallas PopUpWindow donde se muestra la información de los elementos creados por el educador y un PopUpWindow para mostrar la ayuda que contiene el simulador para guiar al educador.
- Paquete games, son los archivos que crean el mini-juego personalizado con los datos introducidos por el educador.
- Paquete js, en el archivo core.js se encuentran todas las funciones que controlan y manejan el simulador. Dentro de este fichero hay:
 - Función que crea los arrays en la base de datos, según los tamaños que el educador introduce en el formulario de la Imagen 26.
 - Funciones para crear distintos tipos de menús dinámicos que están situados dentro de los formularios.

- Función que extrae la información de la base de datos en arrays.
- Funciones para guardar la información en la base de datos.
- Funciones para introducir la información de los elementos guardados en la base de datos en las entradas de los formularios de editar y en las pantallas PopUpWindow de ver elemento.
- Funciones para crear, borrar y guardar la información para crear los formularios que desea introducir el educador en su juego.
- Función para guardar los datos de los formularios del simulador del educador.
- Función que crean y guardan las precondiciones de los retos.
- Funciones que escriben los ficheros del juego del educador.
- Funciones para guardar y sacar la puntuación de los juegos de la base de datos.

El módulo lib contiene:

- Paquete assets, en esta carpeta están guardadas todas las imágenes del simulador.
- Paquete stages, define los recursos que forman el simulador.
- Paquete flow, define los eventos y las transiciones entre los estados del simulador.
- Paquete answer y dialog, guardan los textos y diálogos del simulador.
- Fichero conf.xml, encargado de definir los datos de la base de datos y los ficheros que forman parte del simulador.

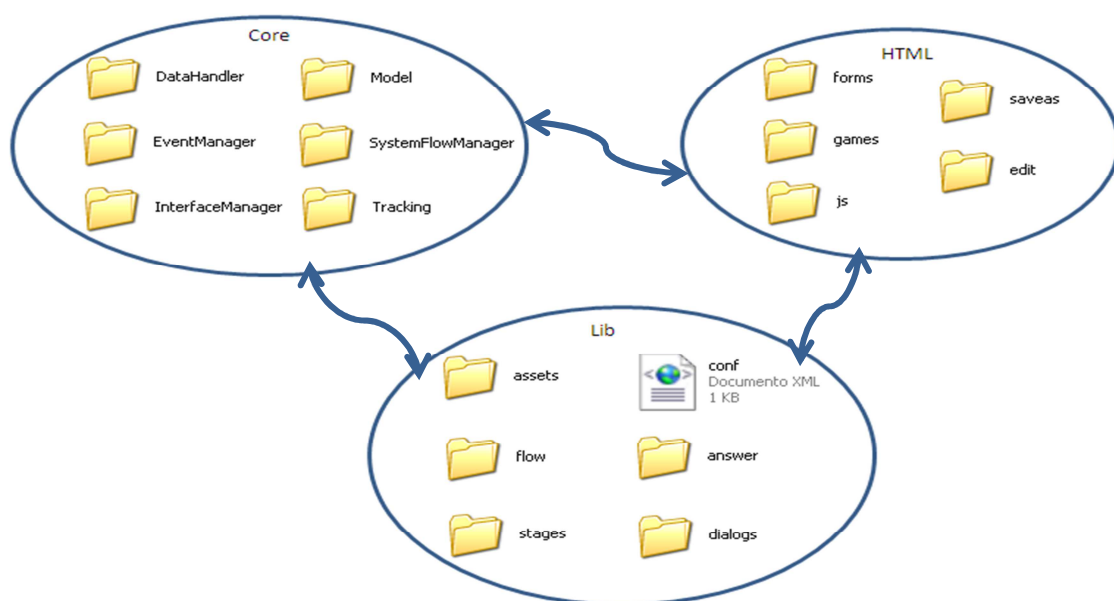


Imagen 29. Módulos del simulador SimGame. Módulos: core, HTML y lib

En la Imagen 30 se muestra un diagrama de flujo del simulador desarrollado en el proyecto para ver el camino que puede seguir el usuario que está creando su juego o simulador.

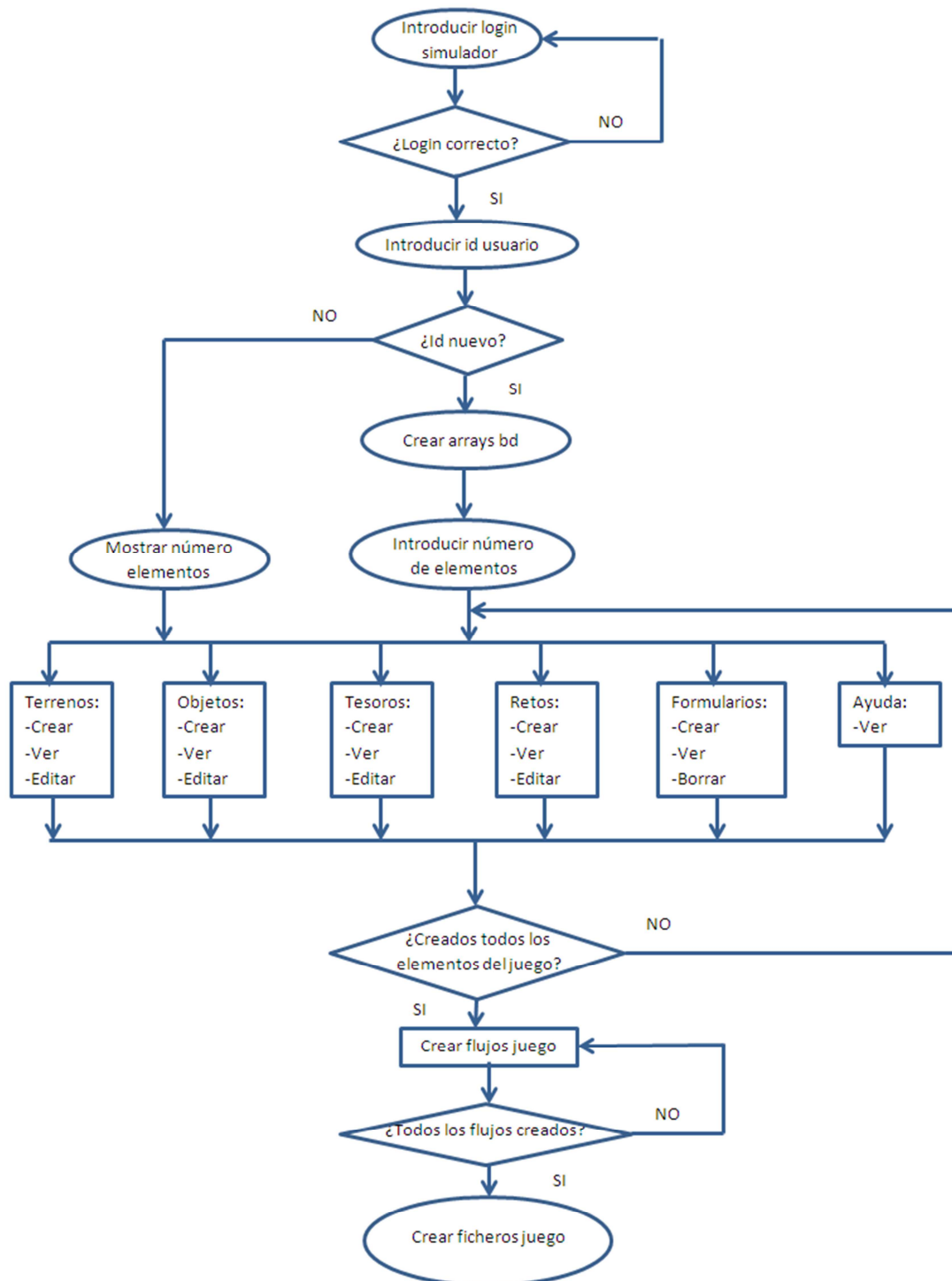


Imagen 30. Diagrama flujo de SimGame

3.7 Diseño. Modelo de datos

El modelo de datos de SimGame, se puede dividir en dos partes: (1) la información propia del simulador que se almacena en ficheros xml y (2) la información propia de los usuarios que utilizan el simulador que se almacena en la base de datos.

Esta decisión de diseño se justifica porque comparativamente el acceso de lectura/escritura es en general más lento para ficheros que para base de datos donde la información se encuentra indexada. También la información propia del simulador cambia menos en el tiempo que la información introducida por los usuarios que utilizan el simulador, ya que requiere una actualización constante durante el proceso de simulación.

Por otra parte, el hecho de tener la información que caracteriza el simulador del usuario en ficheros externos da mucha flexibilidad. Aunque el simulador haya sido creado para usuarios sin conocimientos en programación, esta decisión de diseño facilitaría su modificación por parte de un usuario con avanzados conocimientos de XML y HTML que no necesitaría utilizar la herramienta guiada ni disponer de permisos especiales de acceso a la base de datos, ya que podría modificar el juego modificando el contenido de dichos ficheros.

3.7.1 Ficheros XML y SCXML

SimGame está compuesto por los ficheros XML, (1) escenario donde se crea la interfaz principal del simulador y (2) el fichero de configuración donde se define la base de datos y la estructura del simulador ver Imagen 31. Y el fichero de flujo escrito en SCXML donde se definen las transiciones entre los estados del simulador, Imagen 34 e Imagen 35 para ver fragmentos de estos ficheros.

```
<?xml version="1.0" encoding="utf-8"?>
<config>
  <database host="localhost" port="3306" username="root" pass="" database="simulador" />
  <stage>
    <regions>lib/stages/common/regions.xml</regions>
  </stage>
  <usertracking>true</usertracking>
  <goals>1</goals>
  <flowmanager init="lib/flow/inicial.xml">
    <flowfile src="lib/flow/inicial.xml" name="Inicio"></flowfile>
  </flowmanager>
</config>
```

Imagen 31. Fichero configuración SimGame

Para crear el escenario del simulador lo que hacemos es crear un fichero XML que contiene dos partes diferenciadas (1) definición de los recursos del escenario del simulador y (2) descripción de los recursos de la parte (1), estas dos partes pueden

verse en la Imagen 32 donde se define y describe el elemento fondo y el elemento a) de la Imagen 27. Para ver más código XML consultar el anexo 1.

```

1) <resources>
    <resource id="1">
        <name>Fondo</name>
        <type>ImagePNG</type>
        <src>lib/assets/fondos/fondo.jpg</src>
    </resource>

    <resource id="2"> <!-- Terreno -->
        <name>Tipo</name>
        <type>ActionPoint</type>
        <src>lib/assets/terreno.gif</src>
    </resource>

2) <views>
    <view>
        <name>Default</name>
        <type>Default</type>
        <resources>
            <resource id="1" region="Fondo" /> <!-- Fondo -->

            <resource id="2"> <!-- Terreno -->
                <position x="665" y="120" z="0" zindex="4" />
                <filters>
                    <glow color="0xFDFD4F" blurX="7" blurY="7" style="default" />
                </filters>
                <events>
                    <onOver>true</onOver>
                    <onOut>true</onOut>
                    <onClick>true</onClick>
                </events>
            </resource>

            <resource id="200"> <!-- Panel terreno -->
                <position x="0" y="-45" z="0" zindex="20" /> <!-- Colocar la ventana de las respuestas -->
                <source id="100">
                    <character src="lib/assets/emprendedor.swf" x="-450" y="-80" scaleX="1.85" scaleY="1.85" />
                    <question>¿Que quieres hacer con el terreno?</question>
                    <answer type="respuesta1" answer="Crear terreno" />
                    <answer type="respuesta2" answer="Ver terreno" />
                    <answer type="respuesta3" answer="Editar terreno" />
                </source>
            </resource>
        </resources>
    </view>
</views>

```

Imagen 32. Código XML pantalla principal SimGame

Para crear este mismo fichero se necesita que el usuario rellene los formularios de elementos terrenos, objetos, tesoros y retos que se ven en la Imagen 27. En la Imagen 33 se puede ver un ejemplo de creación del elemento terreno, a) es el formulario necesario para crear el fondo del juego, b) código XML que se crea en la fase de creación de archivos y c) resultado de los pasos a) y b).

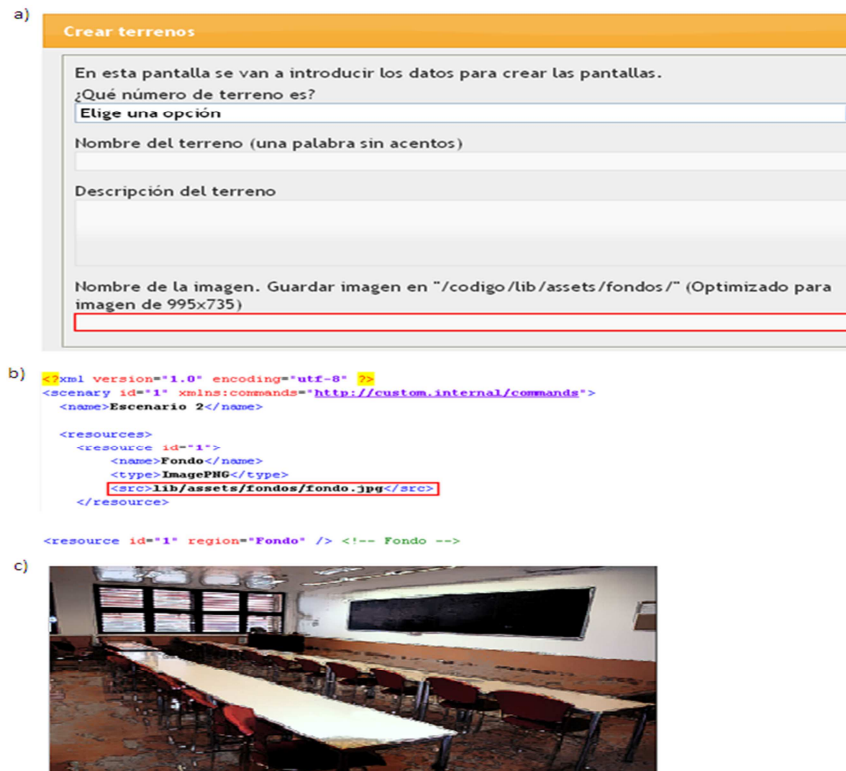


Imagen 33. Ejemplo creación terreno juego. a) formulario, b) código XML, c) resultado

Para crear el flujo del simulador se crea un fichero SCXML que está compuesto de estados, dentro de cada estado hay unas transiciones que hacen activarse ciertos eventos dentro del simulador o pasar a otro estado, un ejemplo de lo mencionado se puede ver en la Imagen 34 para más información sobre el fichero de flujo ver anexo 2.

```
<state id="sec1">
  <state id="sec11">
    <onentry>
      <commands:sendToInterface id="8" command="enable" />
    </onentry>
    <transition event="assetClick 8" target="sec100"><!-- Click juego-->
      <commands:sendToInterface command="external" title="Pantalla de ayuda" url="html/forms/form ayuda.html" width="500" height="540" />
    </transition>
  </state>
```

Imagen 34. Fichero flujo SCXML. Recuadro rojo estado, azul transición y verde evento.

Para crear el fichero de flujo del juego del educador se necesita que el usuario rellene el formularios correspondiente (componente e de la Imagen 27). En la Imagen 35 se puede ver a) el formulario dinámico que se genera dependiendo de los elementos que cree el usuario, b) fragmento de código del flujo que se genera por rellenar este elemento. La correspondencia en colores establece la relación entre información introducida en el formulario y el código que se genera.

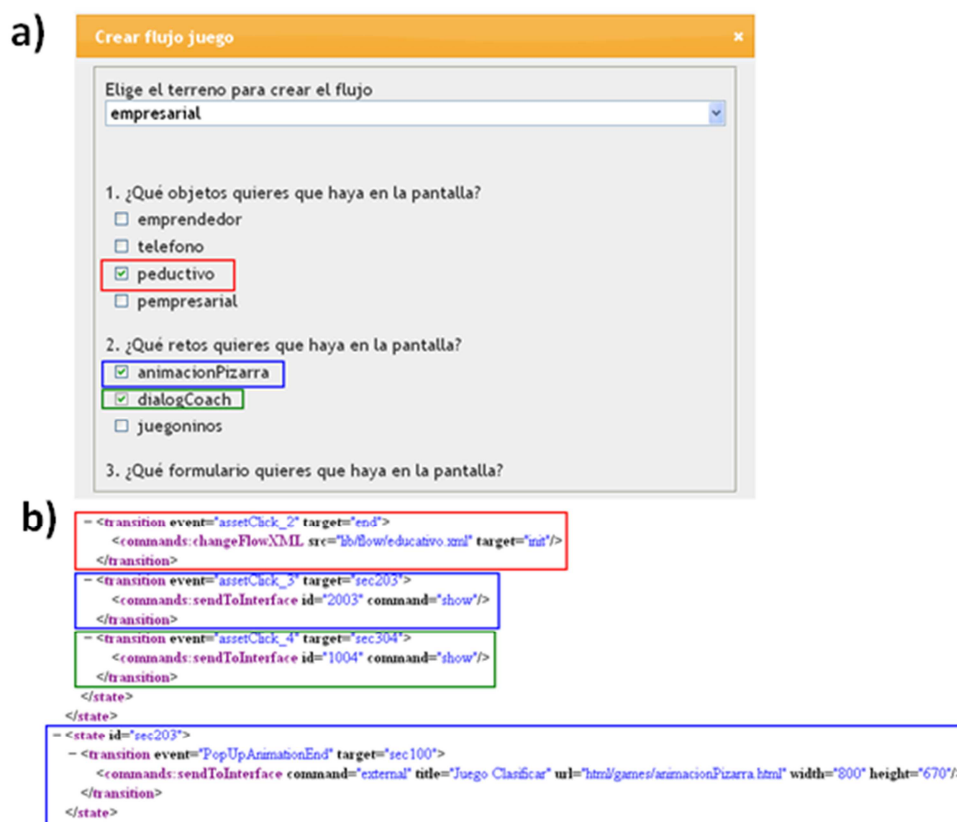


Imagen 35. Ejemplo flujo SCXML. a) formulario presentado en SimGame para introducir el flujo y b) código generado para los ficheros personalizados del usuario.

3.7.2 Base de datos MySQL

La base de datos de SimGame se compone 3 tablas que guardan toda la información del juego, (1) usuario y contraseña para acceder al simulador, (2) seguimiento del usuario en el simulador y (3) arrays de tamaño fijo que guardan la información de los elementos terrenos, objetos, retos y tesoros y arrays de tamaño variable que guardan la información de los formularios de los mini-juegos, el flujo y un identificador generado automáticamente en la fase de creación de ficheros. Todas estas tablas guardan toda la información que genera el usuario durante su interacción con el simulador, pero es la (3) en la que se guarda la información necesaria para crear el juego y es por tanto la que se va a explicar en detalle.

En esta sección se muestran los dos tipos de arrays generados en la base de datos, los arrays de tamaño fijo y los arrays de tamaño dinámico.

La decisión de diseño que se tomó para guardar toda la información en la base de datos fue crear arrays que se guardan dentro de tablas. En un primer momento se pensó en guardar toda la información (nombre, posición, nombre imagen, etc.) de un elemento (terreno, objeto, tesoro y reto) en un array. Esta solución no era óptima a la hora de extraer y editar la información una vez guardada en la base de datos porque la

colocación de la información que guardaba el array dentro del formulario no era automático ya que no coincidía el atributo name del formulario con el nombre del array en la base de datos.

La decisión de diseño que está implementada en el proyecto actualmente es la que nos proporciona más agilidad, ya que se crea un array por cada tipo de información (nombre, posición, nombre imagen, descripción) del elemento (terreno, objeto, tesoro y reto). Para crear el tamaño del array se presenta al usuario el formulario de la Imagen 26. A continuación vamos a crear un terreno de ejemplo para ilustrar de manera visual como se generan los arrays y se guarda la información en ellos, que el usuario introduce a través de los formularios.

En la Imagen 36 se puede ver a) código HTML del formulario y b) arrays creados en la base de datos.

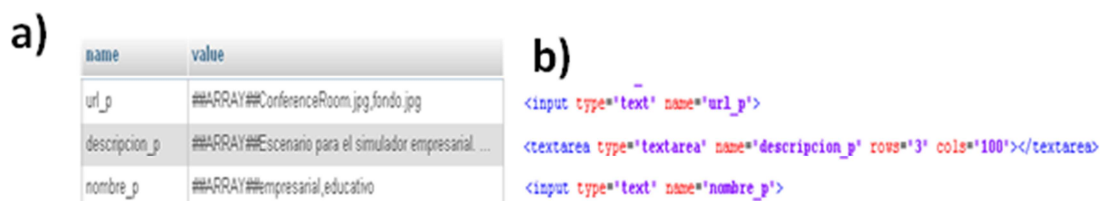


Imagen 36. Ejemplo guardar terreno BD. a) Arrays en la BD, b) código HTML formulario

Para que la acción de guardar la información en la base de datos sea sencilla, con el nombre que hemos declarado en el input del formulario se genera un array con el mismo nombre en la base de datos como se puede ver en la Imagen 36. Para saber la posición en el que se debe de guardar cada dato en el array, en el formulario tenemos un menú desplegable ver Imagen 37 con el número de elementos que se ha introducido en el formulario de la Imagen 26, este menú solo muestra las posiciones de los elementos que faltan por crear.

Para guardar los datos en la posición adecuada al número que se recoge en el campo de la Imagen 37 se le resta uno, debido a que el rango de los índices de los arrays comienza en cero y llega hasta el tamaño introducido menos uno.



Imagen 37. Ejemplo menú elegir número elemento a crear

El procedimiento que se acaba de explicar y que se muestra en Imagen 36 se repite para guardar la información de los objetos, tesoros y en la información de la presentación de los retos.

Del mismo modo, para sacar la información de la base de datos para introducirla en los formularios de editar se crean los inputs con el mismo valor que presenta el atributo name del formulario correspondiente como se ve en la Imagen 38. Así el campo nombre de la base de datos también corresponde con el del formulario y de este modo se simplifica el proceso de rellenar el formulario con los datos previamente introducidos por el usuario.

a)	b)														
<table border="1"> <thead> <tr> <th>name</th><th>value</th></tr> </thead> <tbody> <tr> <td>oy</td><td>##ARRAY##5,375,150,150,0</td></tr> <tr> <td>ox</td><td>##ARRAY##155,535,650,650,0</td></tr> <tr> <td>url_o</td><td>##ARRAY##emprendedor.swf,telephone.png,next.gif,ba...</td></tr> <tr> <td>descripcion_o</td><td>##ARRAY##Guia del simulador,Elemento para llamar a...</td></tr> <tr> <td>pasar_pant</td><td>##ARRAY##no,no,si,si,null</td></tr> <tr> <td>pasar_terreno</td><td>##ARRAY##null,null,educativo,empresarial,null</td></tr> </tbody> </table>	name	value	oy	##ARRAY##5,375,150,150,0	ox	##ARRAY##155,535,650,650,0	url_o	##ARRAY##emprendedor.swf,telephone.png,next.gif,ba...	descripcion_o	##ARRAY##Guia del simulador,Elemento para llamar a...	pasar_pant	##ARRAY##no,no,si,si,null	pasar_terreno	##ARRAY##null,null,educativo,empresarial,null	<pre> <input type="text" id="oy" name="oy"></input> <input type="text" id="ox" name="ox"></input> <input type="text" name="url_o"> <textarea type="text" name="descripcion_o" rows="2" cols="100"></textarea> <input type="radio" name="pasar_pant" id="no" value="no" onclick="crear_pant(1);">No <input type="radio" name="pasar_pant" id="si" value="si" onclick="crear_pant(2);">Si <select name="pasar_terreno"> </pre>
name	value														
oy	##ARRAY##5,375,150,150,0														
ox	##ARRAY##155,535,650,650,0														
url_o	##ARRAY##emprendedor.swf,telephone.png,next.gif,ba...														
descripcion_o	##ARRAY##Guia del simulador,Elemento para llamar a...														
pasar_pant	##ARRAY##no,no,si,si,null														
pasar_terreno	##ARRAY##null,null,educativo,empresarial,null														

Imagen 38. Ejemplo sacar datos objeto BD. a) arrays BD b) inputs formulario editar objeto

Para extraer los datos la única información necesaria es en la posición de cada uno de los datos en el array, para ello se muestra al usuario un menú (Imagen 39) con los elementos que están creados así al pulsar una opción podemos obtener automáticamente la posición requerida.

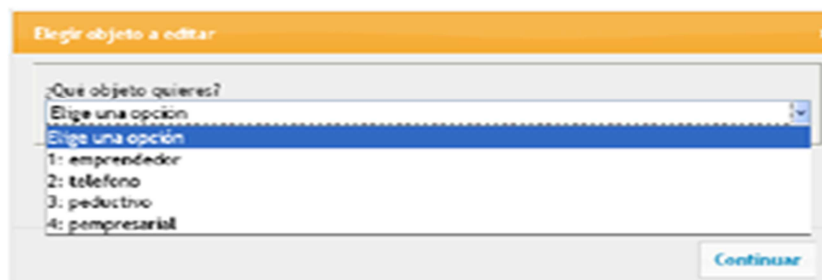


Imagen 39. Ejemplo menú editar elemento objeto para saber posición array

El otro tipo de tablas que se ha nombrado en este apartado son los arrays dinámicos, estas tablas son generadas para guardar la información de los formularios, los datos de los juegos de clasificar y ordenar y la información del flujo del juego del usuario.

En el caso del formulario este array es necesario debido a que es un elemento donde el usuario puede introducir el número de inputs para el formulario que desee y cada input tiene un número también variable de opciones, por tanto no podemos crear un array con un tamaño fijo ni crear varios arrays para guardar la información, ya que no podemos saber cómo va a ser el formulario creado.

Para guardar la información que permite crear posteriormente el fichero del formulario se han creado dos arrays con tamaños variables, uno para guardar el número y tipo de cada input y otro para guardar los textos que quieren mostrar al jugador. Se ha optado por esta solución porque a la hora de crear el formulario resulta más fácil consultar un array con el tipo de input y buscar en la misma posición del otro array los textos que recorrer un único array, ya que tendríamos que poner condiciones para saber cuándo empieza y termina la información de un determinado input. Además se ha creado otro array para llevar un registro de todos los elementos formulario creados por el usuario. En la Imagen 40 se puede ver la correspondencia entre la tabla generada en la base de datos y los campos del formulario que se guardan en cada tabla.

a)

id_form	value
formulario2	MAPARRAY(2),checkboxes,opciones para pulsar1,opc...
formulario2	MAPARRAY(formulario2,ads,ads,ads)
nombre_formulario	MAPARRAY(formulario2,formulario)

b)

Crear formulario

En estas pantallas se van a introducir los datos para crear los formularios.

Nombre del formulario

¿Cuántas entradas quieres?

Eligen el número de entradas del input

¿Que tipo de input quieres?

Tipo input

Introduce la pregunta y los valores de los botones

Imagen 40. Ejemplo crear elemento formulario. a) array creados en la BD, b) formulario mostrado al usuario para crear formulario

Para guardar la información de los retos en el caso que sean mini-juegos de clasificar u ordenar se necesitan estos arrays porque el usuario puede elegir el número de elementos en el juego que necesite además de poder incluir o no imágenes.

Para guardar la información en la base de datos se crean dos arrays, (1) para guardar el tipo de entrada del formulario si es el texto del elemento, la posición o el nombre de la imagen del elemento y (2) para guardar los valores de los textos, posiciones y nombres de las imágenes. Se ha decidido de crear de esta manera, porque a la hora de ver qué porción de código hay que escribir en la fase de creación de los archivos, es más fácil recorrer dos arrays que ir poniendo condiciones especiales para saber qué guarda cada posición del array. En la Imagen 41 se puede ver la correspondencia entre las entradas del formulario mini-juegos y los arrays en la base de datos.

a)

¿Qué número de reto es?

Elige una opción

Elige el número de entradas que quieres en el juego

Número de entradas

Elemento 1

.....

¿Quieres que los elemento sean imágenes?

☐ No ☐ Si

b)

nombres_juego:3

```
##ARRAY##num_input,elementos,posicion,elementos,po
sicion,elementos,posicion,elementos,posicion,elemen
```

array_juego:3

```
##ARRAY##5.hormiga,1.raton,2.perro,3.vaca,4.elefan
te,5.no
```

Imagen 41. Ejemplo crear mini-juego. a) formularios html del reto, b) arrays en la BD

Para guardar la información del flujo del juego en la base de datos se necesitan estos arrays dinámicos porque el usuario puede añadir tantos elementos al terreno como desee, por tanto no sabemos cuál es el tamaño del array hasta que no está creado.

Se crean dos array dinámicos (1) guarda el tipo de elemento que es terreno, objeto, reto o formulario y (2) el nombre del elemento. En la Imagen 42 se puede ver la correspondencia entre los campos del formulario y los elementos guardado en los arrays de la base de datos. Se ha decidido guardar de esta forma porque así sabemos qué código tenemos que introducir en cada momento en la fase de creación de ficheros y no tenemos que insertar condiciones para saber cuándo empieza y termina cada elemento.

a)

name	value
tipo_crea:empresarial	##ARRAY##terreno,objetos,reto,reto
flujo:empresarial	##ARRAY##empresarial,peductivo,animacionPizarra,dialogCoach

b)

Elige el terreno para crear el flujo

empresarial

1. ¿Qué objetos quieres que haya en la pantalla?

☐ emprendedor

☐ telefono

☐ peductivo

☐ pempresarial

2. ¿Qué retos quieres que haya en la pantalla?

☐ animacionPizarra

☐ dialogCoach

☐ juegoninos

3. ¿Qué formulario quieres que haya en la pantalla?

Imagen 42. Ejemplo crear flujo juego. a) arrays BD y b) formulario de flujo

Para terminar con la decisión de diseño de la base de datos se va a explicar como un mismo usuario puede crear varios juegos al mismo tiempo. Lo que tiene que hacer es crear una base de datos en phpMyAdmin ver anexo 4 por cada juego o simulador que desee crear, de esta forma el mismo usuario y contraseña en el simulador para todos los juegos que esté creando.

3.7.3 Ficheros personalizados del juego del usuario

Para crear los archivos del simulador del educador la mejor solución fue crear los ficheros en ventanas a parte de la ventana del simulador, donde se muestra el código y el educador pueda guardar los ficheros manualmente.

Se tomó esta decisión en el diseño porque es la forma más rápida que hay para modificar cualquier archivo, ya que si quiere cambiar un elemento del juego lo único que hay que hacer es volver a guardar las hojas con este elemento y nada más. También facilita a personas con conocimientos de programación poder ver el código generado en tiempo real por si quiere introducir algún cambio. Además de esta forma no se necesitan permisos especiales para escribir en el ordenador y es más rápido la creación de los ficheros.

Para ayudar al usuario a saber en qué parte de la carpeta del código debe de guardar cada hoja, al principio de cada hoja hay un comentario donde se indica qué fichero es y el lugar donde los debe de guardar. Los ficheros que se escriben son de cuatro tipos, escenarios, flujo, ventanas HTML que forman parte de los retos o de formularios y el fichero núcleo del simulador que es el de configuración.

Con la solución que se ha implantado en la generación de los arrays en la base de datos, esta fase de crear los ficheros personalizados se simplifica. Con los arrays del flujo que se han explicado en el apartado anterior sabemos qué tipo de elemento terreno, objeto, reto o formulario es el que debemos de escribir, como también guardamos el nombre podemos buscar en los arrays que guardan el nombre la posición en la que está guardado y con esa posición ya sabemos dónde está toda la información guardada. Ver Imagen 43 para ver ejemplo de cómo están guardados los datos en los arrays de la base de datos, para facilitar la escritura de los ficheros.

a)

```

for(var z=0; z<array_flujo[0].length; z++){
  switch(array_flujo[1][z]){
    case "terreno":
      base_datos(a_terrenos,array_terrenos);
      sacar_pos(array_terrenos[0],ter_el);
      var datos_t=array_terrenos[1][num_el];
      var xmlTerreno = '<resource id="'+(z+1)
      xmlViews+ '<resource id="'+(z+1)+' " r
      id_guardar[id_guardar.length]=(z+1);
      break;
    case "objeto":
      base_datos(a_objetos,array_objetos);
      sacar_pos(array_objetos[0], array_fl
      var datos_o=array_objetos[1][num_el];
      xmlResources+ '<resource id="'+(z+1)+
      xmlViews+ '<resource id="'+(z+1)+' "><
      id_guardar[id_guardar.length]=(z+1);
      break;
    case "reto":
      base_datos(a_retos,array_retos);
      sacar_pos(array_retos[0], array_fluj
      var datos_r=array_retos[1][num_el],a
      xmlResources+ '<resource id="'+(z+1)+
      xmlViews+ '<resource id="'+(z+1)+' "><
      id_guardar[id_guardar.length]=(z+1);
  }
}

```

b)

name	value
tipo_crearempresarial	##ARRAY##terrenoobjeto,reto,reto
flujoempresarial	##ARRAY##empresarial,eductivo,animacionPizarra,di...
idempresarial	##ARRAY##1,3,4

Imagen 43. Ejemplo escribir ficheros personalizados. a) código JavaScript para crear fichero b) arrays flujo guardados en BD

Como los elementos en el código necesitan tener un identificador para saber los eventos que realizan cada uno, a la vez que se crea el código de cada elemento se le va asociando un id que se va guardando un array en la base de datos, como se coloca en la misma posición que el elemento en el array de flujo, sabemos que id corresponde a cada elemento, en la Imagen 43 se ve la correspondencia entre los elementos del array de id y los elementos del arrays que guardan el flujo .

Primero hay que crear los ficheros de cada terreno del juego por separado, estos ficheros son el de escenario, el del flujo y las ventanas html de formularios y mini-juegos y por último se crea el fichero de configuración del juego.

Capítulo 4

Validación y Pruebas

La herramienta desarrollada ha sido probada en diferentes equipos con distintas configuraciones.

El simulador está desarrollado en Flash y puede ser desplegado en una máquina local o en un servidor web. Inicialmente se probó a ejecutar el simulador desplegándolo en la máquina local pero se descartó esta forma ya que hay que descargarse un socket, que escucha los permisos de la política de seguridad de Flash, [4] e iniciarlo a mano para que funcionase Flash, esto dificulta la instalación de la herramienta puesto que obligaría a realizar esta operación en todos los ordenadores desde los que se fuese a crear un simulador. Si se quiere que todos los profesores de un centro puedan crear un simulador para los alumnos sería necesario hacer una operación equivalente para cada uno de sus equipos y el procedimiento a seguir variaría ligeramente en función del equipo, versión del sistema operativo, etc. Por ello finalmente se optó por desplegar el simulador en un servidor web que facilita la instalación y además ofrece una forma de despliegue homogénea independiente del equipo del usuario final. En concreto se decidió la instalación de XAMPP v3 [2] porque además de ofrecer un entorno seguro para la ejecución de los contenidos Flash proporciona la instalación de una base de datos necesaria para guardar la información para crear el juego del usuario y nos facilita también guardar el tracking del seguimiento de usuario.

El simulador ha sido desarrollado en un ordenador portátil con sistema operativo Windows XP (32 bits) y procesador Intel Centrino Duo, pero ha sido probado también en un ordenador de sobremesa con sistema operativo Windows XP (32 bits) con procesador AMDATHLON. Las pruebas se han realizado también en distintos navegadores como se puede ver en la Tabla 15.

Tabla 15. Compatibilidad del simulador SimGame en navegadores

Navegador	Grado de compatibilidad
Mozilla Firefox v.23	Compatible
Chrome v.29	Semi-compatible
Internet Explorer v.8	No compatible

Se realizaron varias fases de pruebas para desarrollar y probar el simulador creado.

1. Se hicieron pruebas para crear simuladores con Simuflow para ver el funcionamiento de la interfaz gráfica, así se vio que elementos se podían introducir en SimGame.
2. Se realizaron pruebas para ver si se generaban correctamente los arrays en la base de datos.
3. Pruebas en SimGame para ver qué información nos faltaba pedir al usuario para crear los elementos, para estas pruebas se crearon a mano varios ficheros XML y SCXML donde se combinaba juegos educativos y elementos de storytellings, y luego se desarrollaron formularios que permitían reproducir estos elementos creados inicialmente a mano
4. Pruebas en SimGame para probar que el simulador que se ha creado es sencillo y se crea un juego de forma rápida, además de utilizar recursos externos (imágenes, vídeos, animaciones) alojados en el propio equipo, y en páginas web.
5. Para la fase de creación de los ficheros personalizados, se realizaron simuladores de prueba que contenían todos los elementos para ir viendo cómo se iba creando el código y así saber si nos faltaba algún dato.
6. Finalmente se creó un juego final que combinaba un simulador empresarial, un juego para niños y un storytelling para comprobar que el simulador funcionaba como multiplataforma. En esta prueba se jugó y se probó el simulador para ver que todos los elementos funcionaban correctamente guardándose la información del jugador y el tracking de usuario.

Cada una de estas pruebas fue seguida de cambios en el simulador y sus resultados son los que motivaron las justificaciones de diseño que se han explicado a lo largo de la memoria. La validación de SimGame con usuarios finales escapa del alcance de este trabajo y queda pendiente como una de las posibles líneas de trabajos futuro

Capítulo 5

Planificación y presupuesto

5.1 Planificación

Desde el comienzo del proyecto se ha fijado una planificación semanal con reuniones presenciales para ir viendo el desarrollo llevado a cabo y definir los siguientes pasos a seguir. Esto ha hecho que el proyecto haya ido respetando fechas hasta la finalización del mismo.

Durante el proceso de desarrollo del proyecto se ha compatibilizado el primer trimestre con asignaturas de la carrera. Y desde su comienzo hasta su fin se ha compatibilizado con unas prácticas en empresa. Esto hace que la dedicación en el desarrollo se explique por horas y no por jornadas laborales de 8 horas.

Hay que destacar que durante el desarrollo del proyecto se produjeron tres parones debido a periodos estivales y periodos de exámenes.

Aunque en la Tabla 16 se puede ver más detallada la evolución, el desarrollo del proyecto ha tenido cuatro fases.

Fase 1. Estudio de la herramienta Simuflow [27].

Este periodo de estudio fue largo ya que es una herramienta nueva con tecnología que no había utilizado antes durante mis estudios y hubo un tiempo grande para analizar su funcionamiento. Se realizaron distintas pruebas para conocer todos los componentes de Simuflow y cómo funciona cada uno, programando distintos tipos de escenarios para que en la fase de desarrollo del simulador fuese lo más rápida posible.

Fase 2. Desarrollo del simulador.

Esta fase es la fundamental ya que fue en la que se estuvo programando el funcionamiento del simulador. Esta fase a su vez se divide en dos partes:

- Base de datos: desarrollo para guardar la información que introduce el usuario para su simulador o juego. Esta parte había que adaptar la base de datos para que se pudiese guardar, sacar y modificar la información fácilmente.

- Creación de la interfaz gráfica: una parte muy importante fue desarrollar la interfaz del simulador, ya que se tenía que ver varias cosas, cómo se presentaba al usuario la información para que el uso del simulador fuese rápido y sencillo, qué datos necesitábamos para crear el juego, cómo era la forma más atractiva de presentar los formularios para que fuese un simulador intuitivo y qué elementos incluye el simulador.

Fase 3. Creación ficheros simulador.

Aunque esta fase ha sido una de las últimas en llevarse a cabo, durante el desarrollo de todo el proyecto se estuvieron analizando diferentes mecanismos y haciendo pruebas para poder crear los archivos que forman el juego del usuario, ya que era uno de los principales objetivos a conseguir.

Durante la fase 2 se estuvieron realizando pruebas creando simuladores de prueba y así conocer los datos que necesitamos que introduzca para crear los ficheros del simulador.

Pero se especifica como una fase distinta ya que hasta que no se completaron el resto de las fases no fue posible diseñar e implementar el mecanismo concreto que permite la generación automática de dichos ficheros, puesto que el modo de hacerlo depende fuertemente de las decisiones tomadas en las fases anteriores, especialmente la forma en que se guarda la información en la base de datos.

Aunque esta fase ha sido la más conflictiva del proyecto, al final cuando ya se tuvo claro la solución final se tardó poco tiempo en desarrollar el código para que funcionase.

Fase 4. Pruebas y corrección errores

La última fase ha sido una fase de pruebas donde se han creado varios simuladores de ejemplo, con esta fase se han visto pequeños detalles que se han ido corrigiendo según surgían, estos detalles sobre todo eran datos que necesitábamos a la hora de crear los ficheros.

En esta fase también se corrigieron pequeños errores del simulador, como textos y estructura de los formularios.

En la Tabla 16 se detalla la dedicación para cada una de las tareas del proyecto.

Tabla 16. Duración de las tareas del proyecto. Al principio con F más un número se especifica en qué fase de las comentadas anteriormente se encuentra cada tarea

Nombre	Inicio	Fin	Horas
F1. Instalación herramientas necesarias y pruebas	19/07/2012	24/07/2012	8
F1. Primer contacto con Simuflow	01/08/2012	07/08/2012	8
F1. Estudio código Simuflow	17/09/2012	01/12/2012	55
F1. Comienzo prototipo simulador	04/12/2013	17/12/2012	20
F2. Creación acciones terrenos y objetos	30/01/2013	12/03/2013	71
F2. Crear acciones presentación retos y tesoros	15/03/2013	01/04/2013	24
F2. Crear menús y solucionar pequeños errores	02/04/2013	10/04/2013	22
F3. Crear ficheros del simulador educador	12/04/2013	29/04/2013	17
F2. Crear acciones retos	18/05/2013	12/06/2013	42
F2. Crear pantalla flujo	13/06/2013	15/06/2013	7
F2. Realizar cambios en retos	17/06/2013	19/06/2013	3
F2. Crear acciones del elemento formulario	20/06/2013	28/06/2013	19
F3. Estudio crear los ficheros del simulador del usuario	01/07/2013	01/07/2013	4
F2. Crear imágenes interfaz gráfica	03/07/2013	03/07/2013	4
F2. Ver juegos del usuario en el simulador	04/07/2013	16/07/2013	14
F2. Crear pantallas guías de usuario	31/07/2013	02/08/2013	9
F2. Cambios en juegos de clasificar y ordenar	05/08/2013	15/08/2013	27
F3. Crear ficheros y crear interfaz para esta acción de crear	17/08/2013	30/08/2013	36
F4. Fase de pruebas y arreglar errores	01/09/2013	10/09/2013	40
Reunión de arranque del proyecto	18/07/2012	18/07/2012	2
Reunión concretar objetivos del proyecto	25/07/2012	25/07/2012	1
Reunión ver pruebas y primeros pasos	21/09/2012	21/09/2012	1
Reunión instalación BD	04/10/2013	04/10/2013	1
Reunión comienzo prototipo	22/11/2012	22/11/2012	1
Reunión ver prototipo y comienzo del simulador	28/01/2013	28/01/2013	1
Reunión ver terrenos y objetos. Sigüientes pasos	14/03/2013	14/03/2013	1
Reunión introducir datos tesoros y retos	21/03/2013	21/03/2013	1
Reunión pantallas introducir datos	11/04/2013	11/04/2013	1
Reunión ver resultados de las pruebas de crear ficheros	25/04/2013	25/04/2013	1
Reunión ver avance proyecto	30/04/2013	30/04/2013	1
Reunión sigüientes pasos	21/05/2013	21/05/2013	2
Reunión ver mini-juegos	10/06/2013	10/06/2013	1
Reunión ver nuevas pantallas de retos y pantalla de flujo	18/06/2013	18/06/2013	1
Reunión ver elemento formulario	02/07/2013	02/07/2013	1
Reunión sigüientes pasos	17/07/2013	17/07/2013	1
Reunión seguimiento del simulador	02/09/2013	02/09/2013	2

Memoria	14/08/2013	23/09/2013	150
TOTAL			597

5.2 Presupuesto

En este apartado se desglosará el presupuesto necesario para el proyecto tanto en coste de personal, como en coste de material.

5.2.1 Coste personal

En la Tabla 17 se detallan los costes de recursos humanos que se han necesitado para el desarrollo del proyecto. No se tendrá en cuenta las jornadas laborales, únicamente se contabilizará las horas que cada persona ha dedicado al desarrollo del trabajo. El jefe del proyecto ha sido M. Carmen Fernández Panadero, el ingeniero técnico y diseñador gráfico ha sido Esther Varas García.

Tabla 17. Coste personal proyecto

	Nº horas	Coste hora	Total
Jefe de proyecto	60	29,65€	1779 €
Ingeniero técnico	560	20 €	11.200 €
Diseñador gráfico	4	20 €	80 €
TOTAL			13.059 €

5.2.2 Coste material

En la Tabla 18 se detallan los costes de material necesarios para desarrollar el proyecto.

Para calcular el coste imputable de cada elemento se ha utilizado la fórmula:

$$\text{Coste imputable} = \left(\frac{\text{dedicación}}{\text{depreciación}} \right) * \text{Coste del producto} * \text{Dedicación en el proyecto}$$

Los valores de dedicación de las herramientas al proyecto y la depreciación de las mismas están calculados en meses.

Hemos establecido una depreciación de 5 años para todas las herramientas.

Tabla 18. Coste de material del proyecto

	Coste	Dedicación	Depreciación	Coste imputable
Ordenador portátil	300 €	10	60	50 €
Windows XP	300 €	10	60	50 €
Notepad ++	0 €	10	60	0 €
XAMPP	0 €	10	60	0 €
Navegador Firefox	0 €	10	60	0 €
Adobe Flash Player	0 €	10	60	0 €

Adobe Photoshop CS6	940 €	10	60	157 €
TOTAL				257 €

5.2.3 Resumen de los costes

En este apartado se calculará el coste total del proyecto teniendo en cuenta el 20% de costes indirectos y el 21% de I.V.A. Tabla 19

Tabla 19. Resumen costes proyecto

Concepto	Importe
Coste de personal	13.059 €
Coste material	257 €
Costes indirectos (20%)	2.663,2 €
Base imponible	15.979,2 €
I.V.A (21%)	3.355,6 €
TOTAL	19.334,8 €

El coste total del proyecto asciende a **diecinueve mil trescientos treinta y cuatro euros con ocho céntimos**.

Capítulo 6

Líneas futuras y conclusiones

6.1 Futuros desarrollos del simulador

En este apartado se van a exponer las líneas futuras del desarrollo del simulador.

6.1.1 Pasar de Flash a HTML5

Uno de los trabajos futuros que se podrían realizar en este simulador es cambiar la tecnología utilizada. Ahora HTML5 es una de las mejores tecnologías web que hay en el mercado por la multitud de facilidades que tiene, ya que no necesita ninguna licencia como ocurre con Flash ya que para cualquier cambio del código ActionScript necesitamos un editor Flash que conlleva pagar una licencia, además de aprender a usar el editor.

Además utilizar HTML5 también tiene una gran ventaja que son los canvas, en este proyecto puede utilizarse para cuando se pide al usuario introducir las coordenadas donde quiere situar una imagen, con canvas podríamos tener una pantalla con la imagen del fondo y tener la posibilidad de colocar la imagen arrastrándolo y que el propio canvas nos diga las coordenadas, lo que facilitaría enormemente el proceso de edición. Actualmente hay un proyecto final de carrera en curso encargado de realizar esta migración.

El uso de HTML permitiría también utilizar el simulador desde dispositivos móviles como tablets o teléfonos móviles.

6.1.2 Crear archivo zip con el juego creado

Actualmente lo que hacemos es crear los archivos que deben de editarse en el simulador y pedirle al usuario que los guarde en su carpeta correspondiente. Esto nos agiliza el cambio de los datos porque sólo necesitamos guardar los archivos que se nos generan.

Una solución más empresarial sería crear un archivo zip con todas las carpetas y todos los archivos que necesita para funcionar además de los archivos que son necesarios editar para personalizar el simulador, así se guardaría todo el conjunto de una sola vez. Descomprimiendo el zip ya tendría el juego.

6.1.3 Agregar lógica para crear videojuegos de aventuras

Crear el código necesario para generar plantillas que guíen al usuario para crear videojuegos de aventuras más avanzados con temáticas específicas.

En el simulador actual ya se pueden introducir las precondiciones que debe de cumplir un jugador para que pueda completar retos y conseguir tesoros, pero se tendría que crear el formulario para las post-condiciones, es decir, que sucede cuando un jugador supera un reto y crear los ficheros en base de estos datos. Con todos estos datos, se podría agregar la funcionalidad de que se vayan activando objetos y pantallas según vas consiguiendo las precondiciones.

6.1.4 Marco regulador.

Para cumplir con el Real Decreto 1494/2007, de 12 de noviembre, por el que se aprueba el Reglamento sobre las condiciones básicas para el acceso de las personas con discapacidad a las tecnologías, productos y servicios relacionados con la sociedad de la información y medios de comunicación social deberíamos de adaptar la herramienta incluyendo.

- Atributo alt a todas las imágenes dentro del simulador donde se describiese la función de cada una.
- La posibilidad de mostrar la ayuda con una síntesis de voz para introducir al discapacitado a la herramienta.
- Para el uso del simulador un reconocimiento de voz para que vaya moviéndose a través del habla.

6.2 Conclusiones

Aunque en sus inicios se había pensado crear un simulador más específico, poco a poco con el desarrollo del proyecto se vio la posibilidad de crear otras funcionalidades creando una herramienta mucho más completa.

A la conclusión de este proyecto se ha construido una herramienta sencilla, rápida y fácil de usar, multifuncional y sobre todo una herramienta que crea juegos y simuladores personalizados sin tener nociones de programación.

Con este proyecto hemos cubierto una necesidad en el mercado que está bajo demanda y en auge.

Se han conseguido todos los resultados previstos y ha sido muy enriquecedor a nivel personal y profesional.

Con el desarrollo de este proyecto, he aprendido tecnologías que no había aprendido durante la carrera como PHP, XML y SCXML. Y esto ha hecho que mis conocimientos sobre tecnologías web hayan aumentado considerablemente, ya que además de aprender nuevas tecnologías, he podido aumentar mi experiencia y mis conocimientos sobre las tecnologías que ya conocía y con las que había trabajado como HTML, JavaScript, Apache y MySQL.

Mi trabajo en la web de Vodafone España también me ha ayudado a desarrollar el proyecto, ya que he podido asistir a estudios de clientes sobre la página, lo que me ha hecho plantearme la forma de pedir la información al usuario. Además también he podido ver con más claridad qué es lo que desea el cliente ante un producto. También el equipo del que formo parte en Vodafone ha mostrado interés en evaluar la herramienta una vez presentado el proyecto para analizar su posible aplicación a algunos de sus proyectos en curso.

Anexos

Anexo 1. Escenario creado con XML

```

<?xml version="1.0" encoding="utf-8" ?>
<scenario id="1" xmlns:commands="http://custom.internal/commands">
  <name>Escenario 2</name>

  <resources>
    <resource id="1">
      <name>Fondo</name>
      <type>ImagePNG</type>
      <src>lib/assets/fondos/fondo.jpg</src>
    </resource>

    <resource id="2"> <!-- Terreno -->
      <name>Tipo</name>
      <type>ActionPoint</type>
      <src>lib/assets/terreno.gif</src>
    </resource>

    <resource id="3"> <!-- Objeto -->
      <name>Tipo</name>
      <type>ActionPoint</type>
      <src>lib/assets/objetos.gif</src>
    </resource>

    <resource id="4"> <!-- Tesoro -->
      <name>Tipo</name>
      <type>ActionPoint</type>
      <src>lib/assets/recompensa.gif</src>
    </resource>

    ...

  </resources>

  <views>
    <view>
      <name>Default</name>
      <type>Default</type>
      <resources>
        <resource id="1" region="Fondo" /> <!-- Fondo -->

        <resource id="2"> <!-- Terreno -->
          <position x="665" y="120" z="0" zindex="4" />
          <filters>
            <glow color="0xFDFD4F" blurX="7" blurY="7" style="default" />
          </filters>
          <events>
            <onOver>true</onOver>
            <onOut>true</onOut>
            <onClick>true</onClick>
          </events>
        </resource>

        <resource id="200"> <!-- Panel terreno -->
          <position x="0" y="-45" z="0" zindex="20" /><!-- Colocar la ventana de las respuestas-->
          <source id="100">
            <character src="lib/assets/emprendedor.swf" x="-450" y="-80" scaleX="1.85" scaleY="1.85" />
            <question>¿Que quieres hacer con el terreno?</question>
            <answer type="respuesta1" answer="Crear terreno" />
            <answer type="respuesta2" answer="Ver terreno"/>
            <answer type="respuesta3" answer="Editar terreno"/>
          </source>
        </resource>

        ...

      </resources>
    </view>
  </views>
</scenario>

```

Anexo 2. Fichero flujo creado con SCXML

```
<scxml xmlns="http://www.w3.org/2005/07/scxml" xmlns:commands="http://custom.internal/commands" version="1.0" initial="init" >
  <state id="init"> <!-- Ventana donde pide el nombre y los apellidos-->
    <onentry>
      <script>trace(' [SCXML] Bienvenidos al Escenario 2');</script>
      <commands:changeStage src="lib/stages/inicial.xml" />
    </onentry>
    <transition event="StageLoaded" target="sec0"><!-- Pantalla login y datos personales-->
      <commands:sendToInterface id="2" command="disable" />
      <commands:sendToInterface id="3" command="disable" />
      <commands:sendToInterface id="4" command="disable" />
      <commands:sendToInterface id="5" command="disable" />
      <commands:sendToInterface id="6" command="disable" />
      <commands:sendToInterface id="8" command="disable" />
      <commands:sendToInterface id="9" command="disable" />
      <commands:sendToInterface id="5000" command="disable" />
      <commands:sendToInterface id="6000" command="disable" />
      <commands:sendToInterface command="external" title="Datos de identificación" url="html/forms/inicial.html" width="500" height="400" />
    </transition>
  </state>

  <state id="sec0">
    <transition event="closedWindow" target="sec1"><!-- Numero de elementos-->
      <commands:sendToInterface command="external" title="Información" url="html/saveas/intro.html" width="500" height="500" />
    </transition>
  </state>

  <state id="sec1">
    <state id="sec11">
      <onentry>
        <commands:sendToInterface id="8" command="enable" />
      </onentry>
      <transition event="assetClick_8" target="sec100"> <!-- Click juego-->
        <commands:sendToInterface command="external" title="Pantalla de ayuda" url="html/forms/form_ayuda.html" width="500" height="500" />
      </transition>
    </state>
  </state>

  <state id="sec10">
    <onentry>
      <commands:sendToInterface id="2" command="enable" />
      <commands:sendToInterface id="3" command="enable" />
      <commands:sendToInterface id="4" command="enable" />
      <commands:sendToInterface id="5" command="enable" />
      <commands:sendToInterface id="6" command="enable" />
      <commands:sendToInterface id="9" command="enable" />
      <commands:sendToInterface id="5000" command="enable" />
      <commands:sendToInterface id="6000" command="enable" />
    </onentry>

    <!-- ////////////////////////////////////// CLICKS////////////////////////////////////-->

    <transition event="assetClick_2" target="sec10"> <!-- Click terreno -->
      <commands:sendToInterface id="200" command="show" />
    </transition>

    <transition event="assetClick_3" target="sec10"> <!-- Click objeto -->
      <commands:sendToInterface id="300" command="show" />
    </transition>

    ...

  </state>
  <final id="end"/>
</scxml>
```

Anexo 3. Formulario creado con HTML

```
<html content="text/html; charset=UTF-8">
```

```
<body>
```

```
<form id="dialogform" type="userextra">
```

```
<fieldset>
```

En esta pantalla se van a introducir los datos para crear las pantallas.

```
<label>iQué número de terreno es?</label>
```

```
<select type="select" id="menu_op"></select>
```

```
<label for="nombre_p">Nombre del terreno (una palabra sin acentos) </label>
```

```
<input type="text" name="nombre_p">
```

```
<label for="descripcion_p">Descripción del terreno</label>
```

```
<textarea type="textarea" name="descripcion_p" rows="3" cols="100"></textarea>
```

```
<label for="url_p">Nombre de la imagen. Guardar imagen en "/codigo/lib/assets/fondos/" (Optimizado para imagen de 995x735)</label>
```

```
<input type="text" name="url_p">
```

```
</fieldset>
```

```
</form>
```

```
<div id="buttons" class="ui-dialog-buttonpane ui-widget-content ui-helper-clearfix">
```

```
<button class="ui-state-default ui-corner-all" type="button">Guardar</button>
```

```
</div>
```

```
</body>
```

```
</html>
```


Anexo 4. Guía de usuario

Para poder utilizar el simulador el usuario tendrá que instalar en su ordenador un servidor, se recomienda descargarse la herramienta XAMPP ya que necesitamos la base de datos MySQL y el servidor apache. Esta herramienta integra estas dos funcionalidades y es sencillo de instalar y poner en funcionamiento.

Para usar XAMPP únicamente tendremos que abrir el panel de control y pulsar el botón “Start” de apache y MySQL como vemos en la Imagen 44.

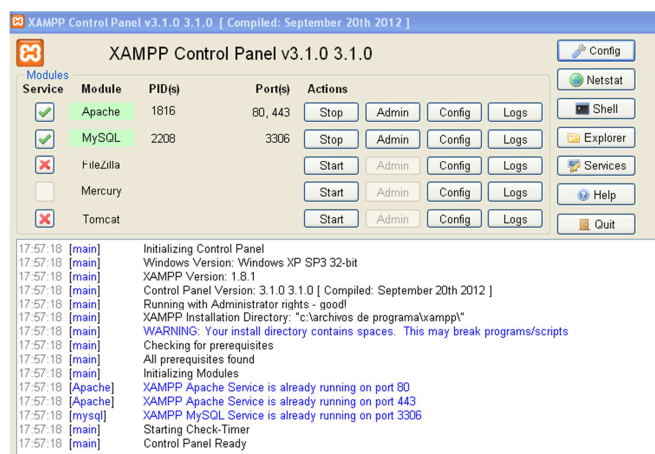


Imagen 44. Guía usuario. Panel control XAMPP

Otro elemento a instalar es el complemento flash de Adobe [1]. Ya que utilizaremos como explorador Firefox Mozilla y no viene integrado este complemento.

Cuando ya tenemos instalado lo necesario en nuestro PC. Tenemos que cargar las tablas que serán necesarias en nuestra base de datos, para ello con un script sencillo que se encuentra en la carpeta del código, vamos al panel de control de phpMyAdmin y realizamos los siguientes pasos:

1. Acceder a través de un navegador a la dirección 127.0.0.1, en el menú derecho de la página que se nos abre clicar sobre phpMyAdmin, y aparecerá el panel de control que aparece en la Imagen 45.

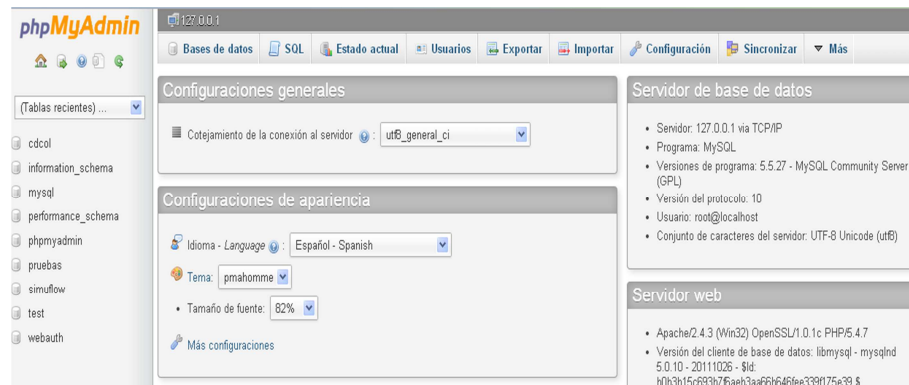


Imagen 45. Guía usuario. panel phpMyAdmin

2. En el menú superior clicar sobre “Base de datos” y rellenar los campos que se pueden ver en la Imagen 46.



Imagen 46. Guía de usuario. Crear base de datos

3. Ir a la tabla creada en el paso 2 a través del menú lateral derecho y en el menú superior la opción de “Importar”, en la pantalla que nos aparece en “Seleccionar archivo” selecciona el fichero bs_simulador.sql que está alojado en la carpeta del código del simulador en el campo de “Conjunto de caracteres del archivo” seleccionar utf-8 como se puede ver en la Imagen 47 y pulsar el botón crear.

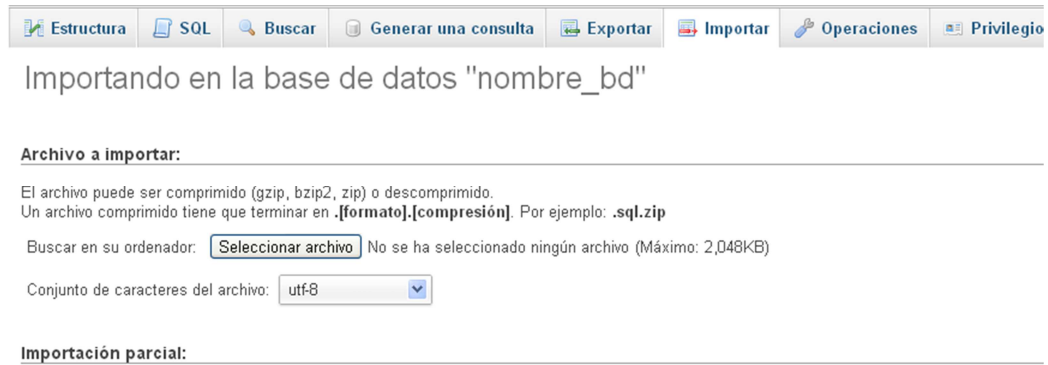


Imagen 47. Guía de usuario. Importar fichero crear tablas BD

4. En el menú lateral derecho Imagen 48 están las cuatro tablas de la base de datos, para crearse un usuario ir a la tabla “users”.



Imagen 48. Guía de usuario. Tablas creadas en la BD

5. En el menú superior clicar sobre la opción “Insertar”, y rellenar los campos que se ven en la Imagen 49 La fila para rellenar la password, se necesita que sea una contraseña encriptada en MID5, online hay multitud de herramientas en las que el usuario puede encriptar su contraseña. Cuando tengamos los tres campos rellenos hay que darle al botón de “Continuar” y ya el usuario tiene un usuario con contraseña para acceder al simulador.

Imagen 49. Guía de usuario. Insertar usuario y contraseña

El resultado de estos pasos es el que se puede apreciar en la Imagen 50

	id	username	password
<div> <div>← T →</div> <div> <div>✎</div> <div>✂</div> <div>✖</div> </div> </div> <div> <div>Editar</div> <div>Copiar</div> <div>Borrar</div> </div>	15	varas	2e6787af57631d5fc4d5cc867de5af3b

Imagen 50. Guía de usuario. Ejemplo usuario creado en la BD

Con todos estos pasos, ya podemos ir al navegador poner la dirección “localhost/nombre_carpeta/index.html” y entrar en el simulador.

A continuación crearemos un simulador empresarial y educativo para demostrar la facilidad del uso y la multifuncionalidad de nuestro simulador.

Para poder crear diferentes juegos y no tener que borrar la base de datos a mano, hemos creado un campo login que puede ser distinto al que usamos para entrar al simulador. Si hemos terminado de crear un juego y queremos crear otro, ponemos otro login al del juego anterior y ya tendremos la base de datos borrada para empezar a introducir los datos del juego nuevo.

Imagen 51. Guía de usuario. Identificar usuario

Como es normal cuando usamos por primera vez una herramienta estamos perdidos. Por eso, para que los usuarios de nuestro simulador no tengan dudas de qué elementos tienen que usar en cada momento, se ha creado una sencilla ayuda. Esta ayuda es mostrada al usuario cuando entra al simulador, y puede acceder a ella en cualquier momento mientras que está creando el juego.

Hemos creado una herramienta sencilla y de rápido uso, para ello hemos pensado tener una única pantalla Imagen 52 con todos los elementos posicionados en ella y así poder acceder a ellos en un click.

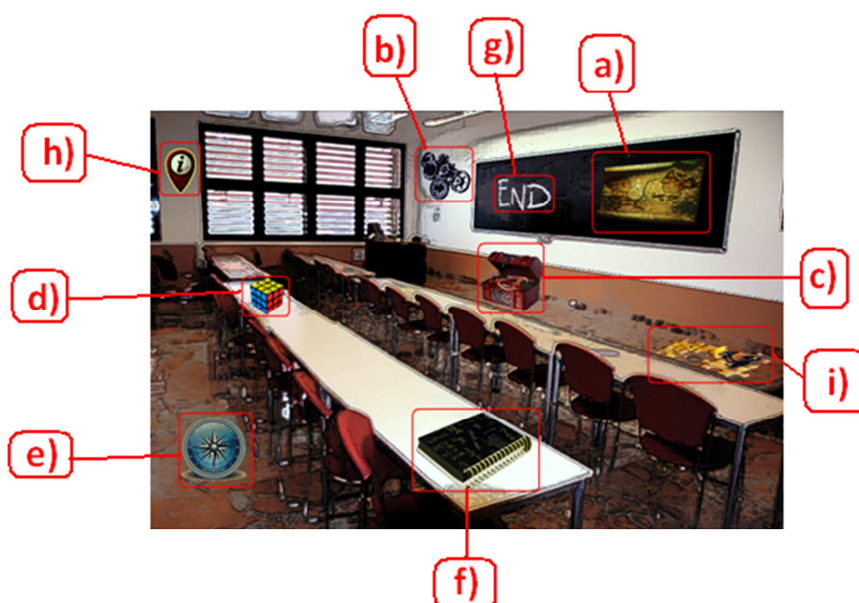


Imagen 52. Guía de usuario. a) terrenos b)objetos c) tesoros d) retos e) flujo
f) formularios g) crear ficheros h) ayuda i) probar mini-juegos

Para empezar a crear el juego vamos al elemento de ayuda Imagen 53 para ver qué idea nos da para crear un juego o por si no nos ha quedado claro qué orden debemos de seguir. Aunque aconsejamos al usuario un orden para crear el juego desde el

principio hasta el final donde se crean los ficheros. Los elementos a) b) c) d) pueden crearse en el orden que queramos ya que podemos editar la información en cualquier momento.



Imagen 53. Guía de usuario. Guía de usuario

Como en todo juego, simulador, historia tenemos un número de pantallas o escenarios y de elementos para interactuar con ellos. En nuestro simulador vamos a utilizar dos escenarios, ya que estamos creando un ejemplo de simulador empresarial y otro escenario para el simulador educativo. Los objetos pueden servirnos como elementos decorativos o en nuestro caso para pasar de una pantalla a otra. Por últimos vamos a crear dos retos para el primer simulador y uno para el educativo.

Esta información que a primera vista parece muy difícil de rellenar, en nuestra herramienta es introducida por el formulario de la Imagen 54. Aunque en un principio pensemos en un número más elevado de objetos en nuestro juego, podemos introducir aquí un valor y como veremos más adelante no es necesario usar todos.

Imagen 54. Guía de usuario. Introducir número de terrenos, objetos, retos y tesoros

Para comenzar a crear el simulador vamos a empezar por lo más básico, los escenarios. Imagen 55 así con la base ya planteada colocar los objetos y retos que lo van a componer es mucho más sencillo.

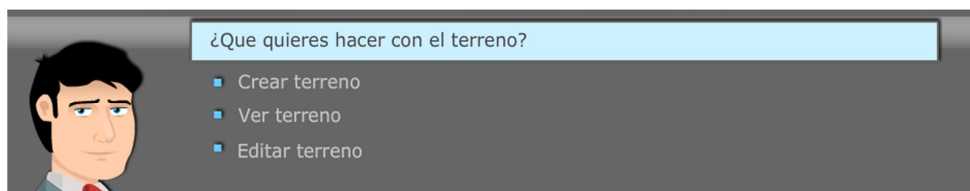


Imagen 55. Guía de usuario. ActionPanel terrenos

Para mostrar lo fácil que es de usar y que no se necesitan recursos muy avanzados, vamos a usar dos imágenes de fondo que ya teníamos en nuestra carpeta. Hay que indicar el número de escenario que es, el nombre con el que vamos a llamar al fichero que más adelante crearemos, la imagen que va a componer el fondo que como se puede apreciar en Imagen 56 solo tenemos que poner la imagen en la carpeta que nos indica y poner el nombre y la extensión en el campo, nos aseguramos de que el usuario no se equivoque al copiar ninguna ruta lo que facilita la introducción de los datos. Por último tenemos un campo de descripción, que nos sirve para saber para qué sirve lo que estamos creando y así, si el usuario no se acuerda qué función realiza este escenario en el simulador puede consultarlo como se muestra en la Imagen 56. Esto hace que no necesariamente se cree todo el simulador de una vez y pueda tardar varios días en terminar el juego.

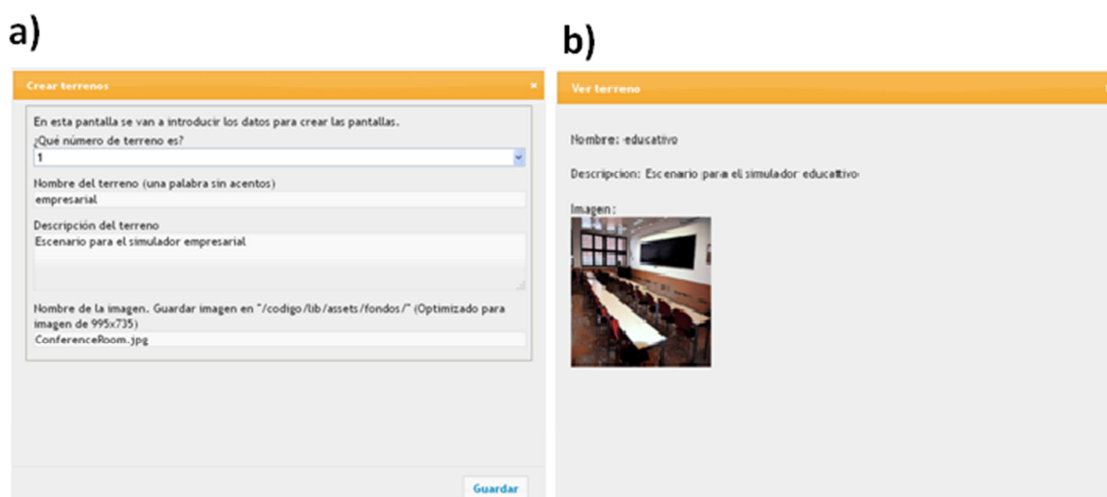


Imagen 56. Guía de usuario. a) Crear terreno b) Ver terreno

Para ver la información de un elemento primero elegimos el elemento que queremos ver en el menú de la Imagen 57, que nos muestra los elementos que hasta ese momento llevamos creados.

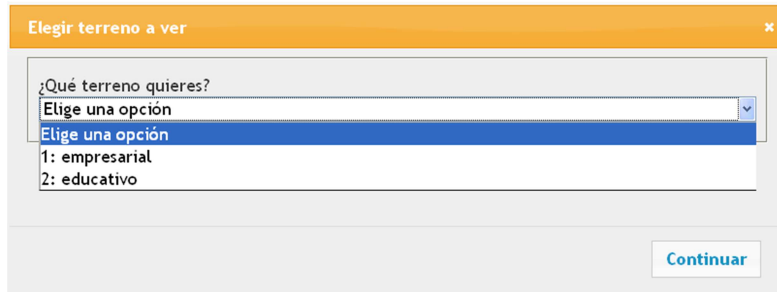


Imagen 57. Guía de usuario. Menú elegir terreno a ver

Si por un casual al ver el escenario Imagen 56, nos damos cuenta que hemos introducido algún dato incorrecto, podemos recuperar los datos y editarlos con la pantalla de la Imagen 58. Esto es una gran ventaja, ya que si cambiamos de imagen podemos reemplazarlo fácilmente y muy rápido.

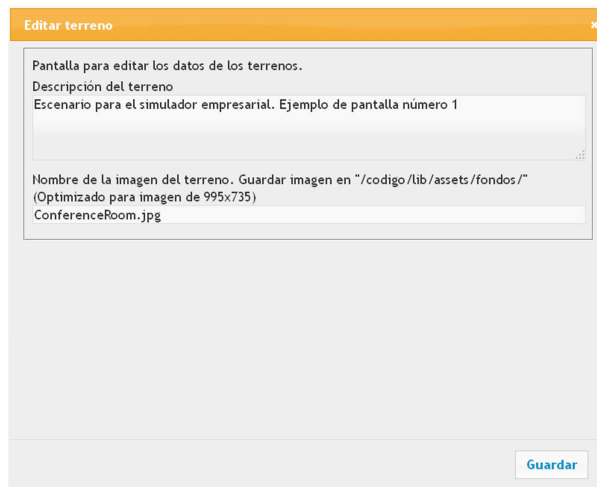


Imagen 58. Guía de usuario. Editar terreno

Para seguir con nuestro simulador, vamos a crear unos objetos que serán unas flechas para indicarle al jugador que puede pasar de una pantalla a otra. Esto lo vamos a hacer rellenando la información de un par de objetos. Como vemos en la Imagen 59, además de los campos que se han comentado para rellenar en el escenario, ahora además, tenemos que introducir la información de la posición en el escenario, lo que nos da una gran flexibilidad al poder poner nuestros objetos en el lugar que nosotros queramos.

Tenemos que indicar si ese objeto sirve para pasar de una pantalla a otra, si la opción es sí como en nuestro caso, tenemos que decirle a qué pantalla queremos que sea la

transición. Aquí se puede ver lo fácil que es el simulador, porque al indicar que si aparece un menú con los escenarios que tenemos creados y sólo con pulsar una opción ya tenemos un dato que al principio puede ser costoso de introducir.

Imagen 59. Guía de usuario. Crear objeto

En el caso de que todavía no hayamos creado el escenario al que queremos que pase, con guardar esta información, podemos más adelante recuperarla cuando tengamos el escenario creado, en la opción de editar Imagen 60.

Imagen 60. Guía de usuario. Editar objeto

Para pedir al usuario que rellene un reto primero hay que explicarle en qué va a consistir el reto o que hay que hacer para superarlo. En el caso de simuladores empresariales, suele ser un ejemplo del ejercicio que a continuación deben de superar.

En nuestro simulador, le ofrecemos tres opciones de presentación de retos, y le indicamos qué datos debe de rellenar dependiendo del reto elegido.

Aunque a priori creamos que es mucha información a introducir, en nuestro simulador con unos pocos datos podemos crear una presentación lo bastante completa para mostrarle al jugador. En el ejemplo que estamos creando lo que vamos a hacer es crear tres retos, lo que primero introducimos es la información de la presentación Imagen 61.

Imagen 61. Guía de usuario. Crear presentación del reto

Como el usuario puede que no se acuerde qué es un popupdialog o un popupanimation, si nos posicionamos encima se nos muestra una imagen de ejemplo como se ve en la Imagen 62.

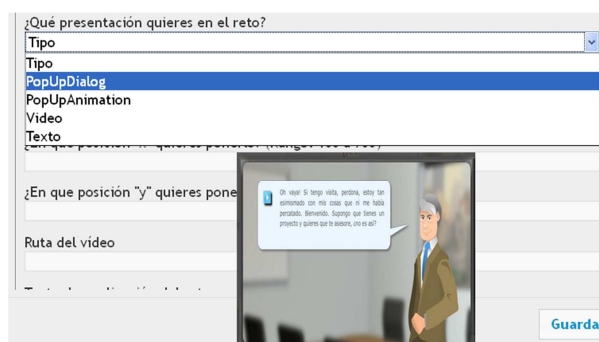


Imagen 62. Guía de usuario. Imágenes de ejemplo en el menú de tipo de presentación

Cuando ya tenemos la presentación creada damos a guardar y la siguiente pantalla sirve para crear el reto. En el simulador hemos metido tres tipos, dos juegos (1) ordenar (2) clasificar y un actionPanel. Igual que pasa en el tipo de presentación, aquí también tenemos la opción de ver que es cada reto Imagen 63.

Imagen 63. Guía de usuario. Elegir tipo de reto

Aunque en un primer momento podamos creer que crear un juego es una actividad complicada, con nuestro simulador en 3 sencillos pasos tenemos un juego creado con un número variable de elemento a gusto del usuario.

En nuestro caso vamos a clasificar tipos de sociedades según el número de socios. Tenemos la opción de que los elementos en vez de palabras puedan ser imágenes, como se aprecia en la Imagen 64, en el caso que sea un simulador educativo con alumnos de corta edad que incluso todavía no sepan leer, se pueda crear un juego con imágenes y el jugador ya pueda jugar sin ningún tipo de problema.

Imagen 64. Guía de usuario. Crear juego de clasificar

Esta funcionalidad es bastante práctica porque además de jugadores de corta edad, puede utilizarse el simulador para personas con discapacidades intelectuales o personas mayores donde el educador necesite crear un juego sencillo y visual.

Para ver cómo nos ha quedado el juego podemos ir al elemento i) Imagen 52 y jugar para ver si necesitamos cambiar algo, o nos damos cuenta que la dificultad no es la deseada. El juego se nos presenta como se le presentará al jugador en el simulador que estamos creando, el ejemplo de nuestro caso se puede ver en la Imagen 65.

Imagen 65. Guía de usuario. Ejemplo juego de clasificar

Para crear un formulario hay que pulsar el elemento f) Imagen 52 primero hay que crear el aspecto que va a presentar el formulario dentro del terreno Imagen 66 después cuando guardamos se nos muestra un formulario para ir eligiendo el número de opciones en la entrada y rellenar los campos que se piden Imagen 66. Para introducir más de una entrada pulsar el botón “Siguiete input” y ya no se desean más entradas en el formulario pulsar “Guardar y terminar”

Imagen 66. Guía de usuario. Crear un elemento formulario a) interfaz formulario b) Crear formulario

Hasta ahora hemos explicado los elementos que da lo mismo el orden en el que los creemos. A partir de ahora, los siguientes pasos si hay que crearlos en orden, ya que son pasos que para crear el juego en el simulador son fáciles, pero a la hora de desarrollar un simulador es necesario tener ya toda la información introducida y claras las ideas para saber qué es lo que va a haber en cada escenario y cuáles van a ser las transiciones entre los escenarios.

Para que el usuario no se olvide de introducir primero qué escenario es el que va a crear el flujo y el contenido, se le mostrará un menú con los escenarios del simulador que ha creado Imagen 67.

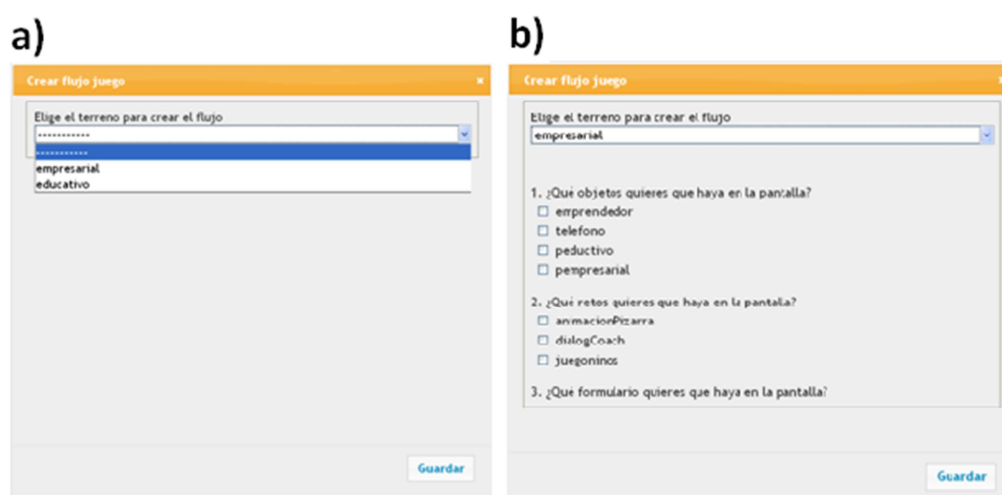


Imagen 67. Guía de usuario. Crear flujo del juego

Cuando ya tenemos seleccionado se nos muestra todos los objetos, retos y formularios que pueden formar parte de ese escenario. Con un simple check podemos seleccionar los que queramos y no tenemos problemas de no acordarnos de los nombres o si no nos acordamos de todos los elementos que hemos creado Imagen 67.

Cuando ya tenemos toda esta información en la base de datos, ya sólo nos quedan crear los archivos que van a formar el simulador. Para guiar al usuario de dónde debe de guardar los ficheros o que es lo primero en qué orden tienen que crearlos, se muestra una pequeña introducción Imagen 68.

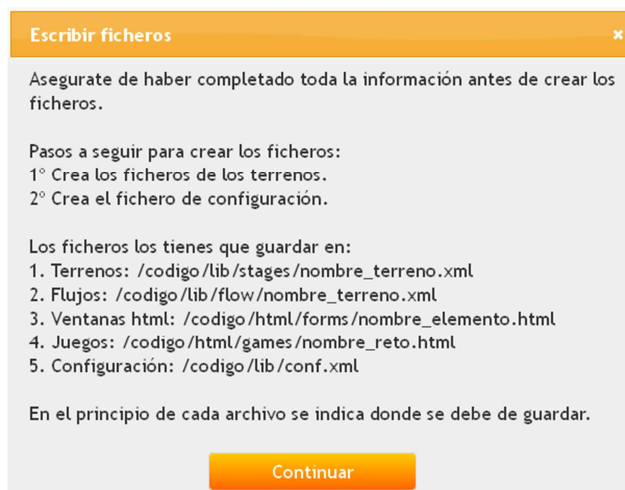


Imagen 68. Guía de usuario. Manual crear ficheros personalizados del juego

Cuando ya hemos leído la explicación, pasamos a crear los ficheros. Para elegir los ficheros a crear es muy sencillo, primero hay que crear el fichero de escenario, flujo y ventanas HTML que forman cada terreno. Primero elegimos en el menú “Terreno” y nos aparecerá otro menú con los terrenos que hemos creado, elegimos el que queremos crear los ficheros y damos al botón de “Escribir”, como vemos en la Imagen 69.

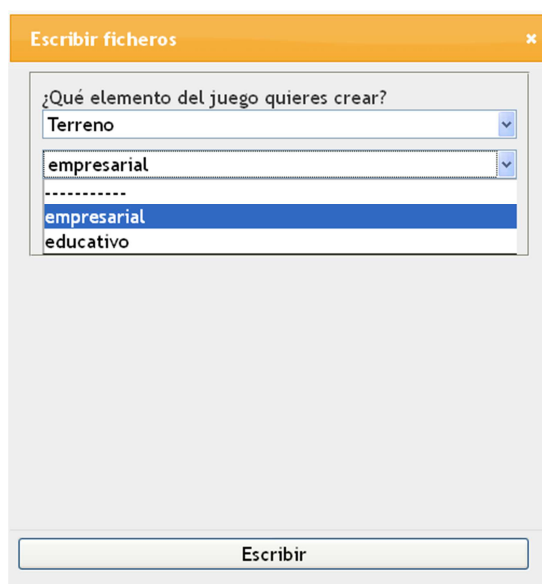


Imagen 69. Guía de usuario. Crear ficheros del terreno del juego

Se abrirá unas ventanas nuevas con el código que tenemos que guardar y se indica al usuario al principio en un comentario verde donde debe de guardarlo por si no se acuerda de la introducción que se le ha mostrado al principio de esta pantalla.

Para guardar el fichero lo único que tiene que hacer es pulsar el botón derecho del ratón y elegir la opción de “Guardar como...”, esta acción es como guardar cualquier archivo o imagen desde Internet ver Imagen 70.



Imagen 70. Guía de usuario. Ejemplo guardar fichero juego

El último paso es crear el fichero de configuración, que es diferente para cada simulador ya que es el núcleo y dice qué escenario componen el simulador. Para crearlo, elegimos en el primer menú la opción de “Configuración” y nos aparecerá la pantalla la Imagen 71, donde primero introducimos el nombre que hemos puesto a la base de datos y luego elegimos cuál va a ser la primera pantalla y ya podemos darle a escribir.

The image shows a dialog box titled "Escribir ficheros" with a close button (X). The text inside the dialog box reads:

Este es el paso final de la creación del juego.

El fichero que se generará ahora, guárdalo en "/codigo/lib" con el nombre conf.xml

¿Cómo has llamado a la base de datos en phpMyAdmin?

¿Cuál es el terreno inicial del juego?

At the bottom of the dialog box, there is a button labeled "Escribir".

Imagen 71. Guía de usuario. Crear fichero de configuración del juego

Hasta aquí hemos visto cómo crear dos tipos de simuladores integrados en un único simulador, de una forma rápida y sencilla.

Ahora para probar el simulador lo único que tenemos que hacer es borrar en el panel de control de la base de datos Imagen 45 el campo que se muestra en la Imagen 72.



Imagen 72. Guía de usuario. Borrar tracking BD

Y en el navegador ponemos la dirección "localhost/nombre_carpeta/index.html" y vemos el resultado Imagen 73.

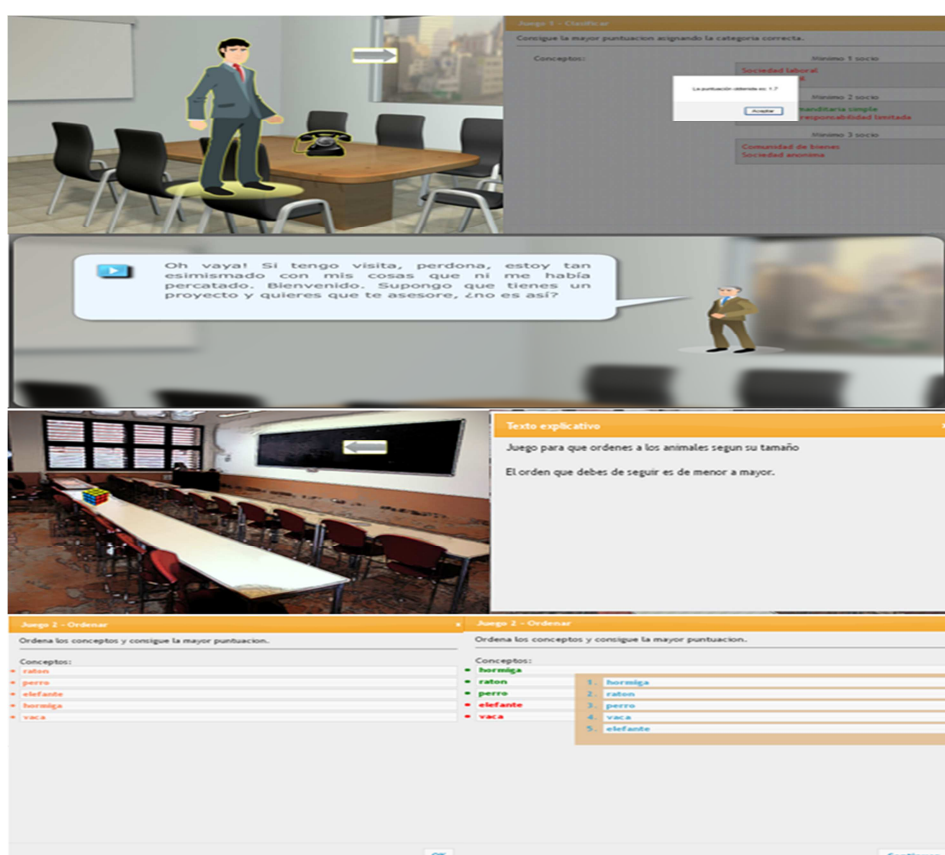


Imagen 73. Guía de usuario. Ejemplo simuladores creados

Como hemos podido comprobar en la Imagen 73, el resultado no es el deseado, pues podemos borrar el contenido del trancking de la base de datos Imagen 72 y volver al simulador para editar los datos que están todos guardados y volver a reescribir los ficheros para que tengan el aspecto que queramos.

Bibliografía

- [1] *Adobe Flash Player*. <http://get.adobe.com/es/flashplayer/> [Julio 2012]
- [2] *Apache Friends*. <http://www.apachefriends.org/en/xampp-windows.html> [Julio 2012]
- [3] *Centro Aragonés de Tecnologías para la Educación*.
<http://www.catedu.es/webcatedu/index.php/descargas/e-adventures> [Septiembre 2013]
- [4] *CodePlex*. <http://socketpolicyfile.codeplex.com/> [Septiembre 2013]
- [5] DICKEY, M. D. (2005). Engaging by Design: How Engagement Strategies in Popular Computer and Video Games Can Inform Instructional Design. . *Educational Technology Research and Development*, 67-83. [Septiembre 2013]
- [6] *e-UCM*. <http://e-ucm.es/> [Septiembre 2012]
- [7] Ferro Soto, C. A. (2009). Ventajas del uso de las TICs en el proceso de enseñanza-aprendizaje desde la óptica de los docentes universitarios españoles. *EDUTEC: Revista electrónica de tecnología educativa*, 29-29. [Septiembre 2013]
- [8] Gómez, L. M. (2004). Entrenamiento basado en la simulación, una herramienta de enseñanza y aprendizaje. *Colomb Anestesiol*, 32-201. [Septiembre 2013]
- [9] Ignacio Rodríguez-García, J. T.-S.-B.-P. (2006). Formación quirúrgica con simuladores en centros de entrenamiento. *Cirugía Española*, 342-348. [Septiembre 2013]
- [10] Kucirkova, N. (s.f.). My story about the 'Our Story'iPhone application. Advice on dealing with the media The confusion surrounding being a 'doctor'The non-motor symptoms of Parkinson's disease,. 26-26. [Septiembre 2013]
- [11] Madrid, C. d.
http://www.madrid.org/cs/Satellite?c=Page&cid=1109182515269&idPadre=1120203321095&pagename=Emprendedores%2FEMPR_Comun%2FDivulgacion%2FEMPR_SimuladorJuegos [Febrero 2013]
- [12] *Manual e-Adventure*. http://e-adventure.e-ucm.es/tutorial/files_ES/eAdventure-v1.3RC-2-Manual_usuario-es_ES.pdf [Septiembre 2013]
- [13] Merino, A. P. (2003). Un simulador de alcance total para la formación de los operarios de sala de control de factorías azucareras. *XXIV Jornadas de Automática*. [Septiembre 2013]
- [14] Ministerio de Educación,C y D
<http://recursostic.educacion.es/fprofesional/simuladores/web/> [Septiembre 2013]
- [15] *Mission Maker*. <http://www.immersiveeducation.com/missionmaker/> [Abril 2013]

- [16] *Mozilla Firefox*. <http://www.mozilla.org/es-ES/firefox/features/> [Julio]
- [17] Pla Sentis, I. (1978). Simuladores de lluvia para el estudio de relaciones suelo-agua bajo agricultura de secano en los trópicos. In *5. Congreso Venezolano de la Ciencia del Suelo. Barquisimeto (Venezuela)*, 5-10. [Septiembre 2013]
- [18] Rabanal, J. M. (2003). Los simuladores médicos en la formación continuada: nuestra experiencia con 553 médicos de urgencias hospitalarios. *Emergencias*, 333-38. [Septiembre 2013]
- [19] SIMmarket. <http://www.simuladores-empresariales.com/metodologia/ObjetivosVentajas.html> [Abril 2013]
- [20] *Simuladores para Formación Profesional*. <http://www.proyectoagrega.es/simuladores/> [Marzo 2013]
- [21] Simumark. <http://simumak.com/es/> [Septiembre 2013]
- [22] *Storybird*. <http://storybird.com/> [Septiembre 2012]
- [23] *Tecnocarreteras*. <http://www.tecnocarreteras.es/web/items/1/739/> [Septiembre 2013]
- [24] *TGC. The Game Creators*. <http://www.thegamecreators.com/> [Septiembre 2013]
- [25] *The Game Maker Company*. <http://www.gamemaker.nl> [Septiembre 2013]
- [26] *Transmedia Storyteller*. <http://www.tstoryteller.com/how-it-works> [Septiembre 2013]
- [27] Vega, D. (2012). Simuflow. Un framework para el desarrollo de simuladores educativos. [Agosto 2013]
- [28] *W3C*. <http://www.w3.org/> [Octubre 2012]
- [29] *X-plane*. <http://www.x-plane.com/es/escritorio/inicio/> [Septiembre 2013]
- [30] ADESE. <http://www.adese.es/> [Septiembre 2013]